










# QCMs, EFORMS, PYTHON, PDFTK ET PYPDF2

## I. Pré-requis

### 📌 Cadre

Afin de profiter au mieux des capacités de  python, ,  et , il faudra commencer par vérifier que les éléments suivants sont bien installés :

- $\LaTeX$  bien évidemment avec les packages  `pythontex` et  `eforms`;
-  python avec le module  `pypdf2` (via conda ou pip);
-  `pdftk` (sa version gratuite suffit).

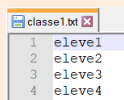
## II. Préparation des fichiers de configuration

### II.1. Les fichiers des « classes »

#### 📌 Intro

Pour commencer, il est conseillé de travailler dans un répertoire temporaire (beaucoup de fichiers temporaires et utiles sont à générer...).


Dans ce répertoire, il y aura, pour commencer les fichiers txt pour chacune des classes à considérer (chaque fichier contiendra les élèves sous la forme d'un pseudo par exemple).



### II.2. Les fichiers de configuration « fdf »

#### >\_ Tuto

L'idée est de préparer le terrain pour créer un fichier QCM par élève, en complétant *par défaut* le champ nomprenom.

Pour cela, il faut déjà configurer les fichiers fdf pour chaque élève, et on va utiliser un script  python.

```

1 import os
2 #Données des classes
3 classepseudos = ['eleve1','eleve2','eleve3','eleve4'] #à compléter
4 classeinoms = ['Élève 1','Élève 2','Élève 3','Élève 4'] #à compléter
5 #puis classe2... classe3...
6 #script de génération des fichiers fdf
7 def genfdf(classe):
8     if not os.path.exists('datasfdf'):
9         os.mkdir('datasfdf')
10    if classe == 'classe1':
11        listepseudos,listeninoms = classepseudos,classeinoms
12    #elif ...
13    n = len(listepseudos)
14    for i in range(n):
15        file = classe + '_' + listepseudos[i] + '.fdf'
16        f = open('datasfdf/'+file, "w")
17        f.write('%PDF-1.2\n')
18        f.write('1 0 obj\n')
19        f.write('<</FDFA<</Fields[<</T(nomprenom)/V('+ listenoms[i] + ')>>]>>\n')
20        f.write('endobj\n')
21        f.write('\n')
22        f.write('trailer\n')
23        f.write('<</Root 1 0 R>>\n')
24        f.write('%EOF')
25    f.close()
26    return
  
```

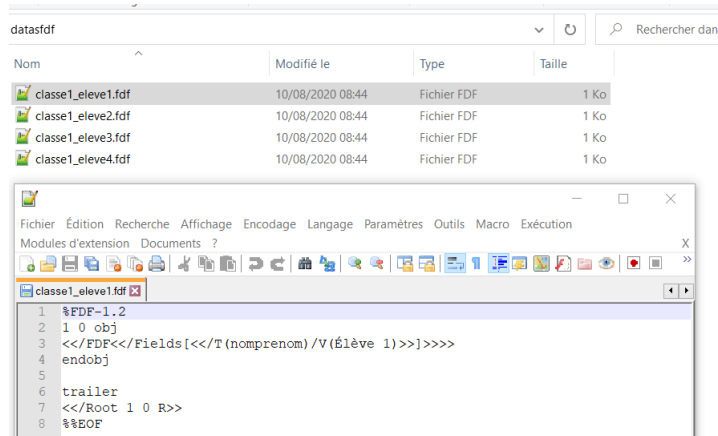
**Illustration**

Le script est à charger (l'exemple est donné pour la classe1) :

- directement dans l'IDE en lançant ensuite la commande `genfdf('classe1');`;
- en utilisant l'invite de commandes et en lançant la commande

```
python -c "import fdfcreate; fdfcreate.genfdf('classe1')"
```

Les fichiers fdf créés (une fois pour toute) ont la structure suivante, qui permettra d'insérer dans le champ <nomprenom> la valeur indiquée :



## II.3. Les codes python et LaTeX pour les réponses placées aléatoirement

### >\_ Tuto

On peut commencer par déclarer les commandes pour créer les boutons via eforms :

```

\usepackage{eforms}
%1 choix parmi 4, on ne peut pas décocher (plus le bouton n°5 pour la non réponse !)
\newcommand\choixqcmu[2]{\radioButton[\V{E}\DV{E}]{#1}{1em}{1em}{#2}}
\newcommand\choixqcmunochoice[1]{\radioButton[\V{E}\DV{E}]{#1}{1em}{1em}{E}}
  
```

[Code LaTeX](#)

On continue par le code générique  python relatif au placement aléatoire :

```

\usepackage{pythontex}
%Code Python pour le mélange des réponses
\begin{pycode}
def qcmalea(a,b,c,d):
    from random import shuffle
    reponses = [a,b,c,d]
    shuffle(reponses)
    return reponses

#éventuellement une version à 3 réponse def qcmalea3()...

#Question1
Q1A = r"\choixqcmu{Q1}{A}~~blablareponseA"
Q1B = r"\choixqcmu{Q1}{B}~~blablareponseB"
Q1C = r"\choixqcmu{Q1}{C}~~blablareponseC"
Q1D = r"\choixqcmu{Q1}{D}~~blablareponseD"
reponsesQ1 = qcmalea(Q1A,Q1B,Q1C,Q1D)

#Question2
Q2A = r"\choixqcmu{Q2}{A}~~blablareponseA"
Q2B = r"\choixqcmu{Q2}{B}~~blablareponseB"
Q2C = r"\choixqcmu{Q2}{C}~~blablareponseC"
Q2D = r"\choixqcmu{Q2}{D}~~blablareponseD"
reponsesQ2 = qcmalea(Q2A,Q2B,Q2C,Q2D)

#etc...
\end{pycode}
  
```

[Code LaTeX](#)

Et de ce fait le code  $\LaTeX$  pour générer le champ `<nomprenom>` et (par exemple) le tableau des réponses avec cases à cocher :

```
%La gestion du champ <nomprenom>
{\Large NOM Prénom : }~{\textField[\V{À remplir}\BC{0 0 0}\textSize{12}]{nomprenom}{12cm}{20pt}}

%Un exemple de question avec les réponses mélangées :
BlaBlaQuestion1

\begin{center}
\renewcommand{\arraystretch}{1.75}
\begin{tabularx}{\linewidth}{|X|X|}
\hline
\py{reponsesQ1[0]} & \py{reponsesQ1[1]} \\ \hline
\py{reponsesQ1[2]} & \py{reponsesQ1[3]} \\ \hline
\multicolumn{2}{c}{\choixqcmunochoice{Q1}~~Je ne préfère pas répondre\dots} \\
\end{tabularx}
\end{center}
```

Code  $\LaTeX$ 

### Illustration

Avec le logiciel Adobe Acrobat Reader (pour les autres il faut vérifier que les cases se cochent bien et mettent à jour les champs!) on obtient le rendu suivant :

NOM Prénom :

**Exercice 3 (COMMUN) ..... (4 points)**  
*Cet exercice est un questionnaire à choix multiples. Pour chacune des questions suivantes, une seule des quatre réponses proposées est exacte. Une bonne réponse rapporte un point. Une réponse fautive, une réponse multiple ou l'absence de réponse à une question ne rapporte ni n'enlève de point. Pour répondre, indiquer sur la copie le numéro de la question et la lettre de la réponse choisie. Aucune justification n'est demandée.*

1. BlaBlaQuestion1

<input type="checkbox"/> blaBlaréponseD	<input type="checkbox"/> blaBlaréponseB
<input type="checkbox"/> blaBlaréponseC	<input type="checkbox"/> blaBlaréponseA

Je ne préfère pas répondre...

2. BlaBlaQuestion2

<input type="checkbox"/> blaBlaréponseA	<input type="checkbox"/> blaBlaréponseC
<input type="checkbox"/> blaBlaréponseD	<input type="checkbox"/> blaBlaréponseB

Je ne préfère pas répondre...

## III. Génération des fichiers élèves et analyse des résultats

### III.1. Fichiers élèves

#### >\_ Tuto

On peut maintenant générer les fichiers élèves à l'aide du script `generateqcmelèves.bat` qui est placé dans le répertoire contenant le fichier `tex` à « dupliquer » :

```
set fichier=%1
set classe=%2
for /F %%x in (%classe%.txt) DO (
copy %fichier%.tex %fichier%_%%x.tex
pdflatex.exe --shell-escape -synctex=0 -interaction=batchmode %fichier%_%%x.tex
pythontex --rerun=always %fichier%_%%x.tex
pdflatex.exe --shell-escape -synctex=0 -interaction=batchmode %fichier%_%%x.tex
del %fichier%_%%x.tex
)
```

&gt;\_ Code Shell

On lance un terminal (ou une invite de commandes) et on lance la commande :

```
cmd generateqcmelèves.bat <nomfichiersansextension> <nomclasse>.
```

### Illustration

On obtient de ce fait un fichier par élève :

- qcmeforms.pdf
- qcmeforms.tex
- qcmeforms\_eleve1.pdf
- qcmeforms\_eleve2.pdf
- qcmeforms\_eleve3.pdf
- qcmeforms\_eleve4.pdf

**>\_ Tuto**

On peut maintenant insérer les données <nomprenom> dans chaque fichier élève à l'aide d'un fichier createqcms.bat qui va chercher les données fdf et les insérer dans les fichiers pdf :

```
set /p fichier="Nom du fichier pdf : "
set file=%fichier:.pdf=%
set /p niv="Choix de la classe : "
set /p dest="Masque du fichier de sortie : "
mkdir out
for /F %e in (%niv%.txt) DO (
    pdftk %file%_%e.pdf fill_form datasfdf\%niv%_%e.fdf^
    output out\%niv%_%dest%_%e.pdf need_appearances)
```

**>\_ Code Shell**

On lance une invite de commandes et on lance la commande `createqcms.bat` et on répond aux questions.

**Illustration**

On obtient désormais un fichier personnalisé par élève :


```
classe1_qcm01essai_eleve1.pdf
classe1_qcm01essai_eleve2.pdf
classe1_qcm01essai_eleve3.pdf
classe1_qcm01essai_eleve4.pdf
```

**III.2. Analyse des résultats****>\_ Tuto**

Une fois les fichiers remplis par les élèves et récupérés, on peut lancer l'analyse avec un script `python`.

```
1 import PyPDF2, glob, sys, os
2
3 def listereponses(nb,fichier):
4     freponse = PyPDF2.PdfFileReader(fichier)
5     reponses = freponse.getFields()
6     liste_reponses = ''
7     nom_copie = reponses['nomprenom']['/V']
8     for i in range(1,nb+1):
9         num = 'Q'+str(i)
10        rep = reponses[num]['/V'][1:]
11        if rep == 'E':
12            liste_reponses = liste_reponses + '-'
13        else :
14            liste_reponses = liste_reponses + rep
15        return nom_copie,liste_reponses
16
17 def note(reponses,bonnesreponses,ptok=1,ptnok=0):
18     note = 0
19     for i in range(len(bonnesreponses)):
20         if reponses[i][i] == '-':
21             note += 0
22         elif reponses[1][i] == bonnesreponses[i]:
23             note += ptok
24         else :
25             note += ptnok
26     return reponses[0],reponses[1],note
27
28 def bilan(bonnesreponses,ptok=1,ptnok=0):
29     nbquestions = len(bonnesreponses)
30     classe = [f for f in glob.glob("*.pdf")]
31     f = open("bilan.txt", "w")
32     f.write('fichier;nom;réponses;note\n')
33     for fichier in classe :
34         res = note(listereponses(nbquestions,fichier),bonnesreponses,ptok,ptnok)
35         ligne = fichier + ";" + res[0] + ";" + res[1] + ";" + str(res[2])
36         f.write(ligne + '\n')
37         print(f"fichier={fichier} ; nom={res[0]} ; réponses={res[1]} ; note={res[2]}")
38     f.close()
```

**Code Python**

On peut maintenant exécuter le script  python :

- directement dans l'IDE avec la commande `bilan('<bonnesréponses>',ptok,ptnok);`
- via l'une invite de commandes dans le répertoire contenant les fichiers récupérés, grâce à la commande `python -c "import recupdatapdf ; recupdatapdf.bilan('<bonnesréponses>',ptok,ptnok)".`

Le fichier réponse est ainsi créé et affiché.


### Illustration

Ci-dessous un test avec :

- des copies de 4 élèves remplies,
- une analyse avec 1 point par bonne réponse et 0 sinon ;
- une autre avec 1 point par bonne réponse et  $-0.5$  sinon.

```
python -c "import recupdatapdf ; recupdatapdf.bilan('ABCD',1,0)"
fichier=classe1_qcm0lessai_eleve1.pdf ; nom=Élève 1 ; réponses=ABCD ; note=4
fichier=classe1_qcm0lessai_eleve2.pdf ; nom=Élève 2 ; réponses=BBBB ; note=1
fichier=classe1_qcm0lessai_eleve3.pdf ; nom=Élève 3 ; réponses=-CBA ; note=0
fichier=classe1_qcm0lessai_eleve4.pdf ; nom=Élève 4 ; réponses=---D ; note=1

python -c "import recupdatapdf ; recupdatapdf.bilan('BBBB',1,-0.5)"
fichier=classe1_qcm0lessai_eleve1.pdf ; nom=Élève 1 ; réponses=ABCD ; note=-0.5
fichier=classe1_qcm0lessai_eleve2.pdf ; nom=Élève 2 ; réponses=BBBB ; note=4
fichier=classe1_qcm0lessai_eleve3.pdf ; nom=Élève 3 ; réponses=-CBA ; note=0.0
fichier=classe1_qcm0lessai_eleve4.pdf ; nom=Élève 4 ; réponses=---D ; note=-0.5
```



```
*Sans titre - Bloc-notes
Fichier Edition Format Affichage Aide
E1 : ABCD
E2 : BBBB
E3 : -CBA
E4 : ---D

Bonnes réponses : ABCD / 1pt bonne réponse / 0pt sinon
Bonnes réponses : BBBB / 1pt bonne réponse / -0.5pt sinon
```