



HAL
open science

PyCloze: Questions cloze individualisées générées en Python pour les tests Moodle

Christian Ghiaus, Frank Tillenkamp

► To cite this version:

Christian Ghiaus, Frank Tillenkamp. PyCloze: Questions cloze individualisées générées en Python pour les tests Moodle. 7e Colloque Pédagogie et Formation, INSA Hauts-de-France, May 2021, Valenciennes, France. hal-04413467

HAL Id: hal-04413467

<https://hal.science/hal-04413467v1>

Submitted on 23 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



PyCloze

Questions *cloze* individualisées générées en Python pour les tests Moodle

Christian GHIAUS^{*1}, Frank TILLENKAMP²

¹INSA Lyon, <https://www.insa-lyon.fr>, France

²ZHAW, <https://www.zhaw.ch>, Suisse

RESUME

PyCloze est une procédure qui utilise un module Python pour obtenir des questions avec des réponses intégrées pour les tests organisés sur la plateforme d'apprentissage en ligne Moodle. Pour son utilisation, on a besoin de connaissance de base de Python 3 et de Moodle 3. La procédure permet de générer un fichier .xml contenant un ensemble de questions, chacune avec des jeux de données entrées-sorties différents. Ce fichier est importé dans une catégorie de la banque de questions de Moodle. Les questions des tests Moodle sont choisies aléatoirement pour chaque étudiant. Comme Moodle peut être accédé depuis un téléphone portable, une tablette ou un ordinateur, les tests peuvent être utilisés pendant les cours magistraux, les travaux dirigés et les examens, que ce soit en présentiel ou en distanciel. PyCloze est disponible en libre accès (licences MIT et CC-BY) sur [zenodo.org](https://zenodo.org/doi/10.5281/zenodo.4101667) (DOI 10.5281/zenodo.4101667).

* christian.ghiaus@insa-lyon

1. INTRODUCTION

Un problème dans les tests faits en utilisant la plateforme Moodle est d'individualiser les questions, c. à d. de générer des versions multiples d'une question à partir de valeurs numériques différentes. Une solution serait d'utiliser une [question calculée](#) qui nécessite les étapes suivantes :

- **Écrire un texte avec des variables *jokers***, encadrées par des accolades, qui seront tirées au hasard parmi un jeu de données prédéfini correspondant à des valeurs possibles. Par exemple : `Quelle est la surface d'un carré de côté {a}`.
- **Créer une fonction qui résolve le problème** qui contient des variables *jokers* apparaissant dans le texte. Un assemble de fonctions mathématiques est disponible en Moodle. Par exemple : `pow({a}, 2)`. Une erreur acceptable est également indiquée.
- **Définir la plage de valeurs pour les variables *jokers***, par exemple 10 valeurs entre 10 et 100.

Une [question calculée](#) permet des formules mathématiques (à partir de fonctions), mais pas des algorithmes, comme ceux qui peuvent être implémentés avec des langages de programmation.

Une [question cloze](#) à réponses intégrées permet de réaliser un test avec des trous dans lesquels les étudiants insèrent les réponses (Moodle, Question cloze à réponses intégrées, 2020). Ces réponses peuvent être numériques, choix multiple et/ou texte court (Figure 1). Mais il n'est pas possible d'individualiser les problèmes en utilisant des variables *jokers* et des fonctions pour le calcul des réponses.

John Smith has 3 children: **Antoinette**, age **10.2**, **Lawrence**, age **12.9**, **Sebastian**, age **9.7**.

The mean age of the children is $\mu =$ (± 0.5).

The eldest child is : .

The child with the longest name is .

Figure 1 Exemple d'une question à réponses intégrées. Les réponses sont de type : numérique, choix multiple et texte court. Une question cloze ne permet pas d'écrire des variables *jokers* (soulignées en jaune).

[PyCloze](#) représente une solution à ces limitations de Moodle. La procédure [PyCloze](#) et les modules [MoodleCloze.py](#) et [MdlClz.py](#) permettent à une [question cloze](#) un comportement similaire à une [question calculée](#), mais avec la flexibilité et la puissance du langage de programmation Python. Pratiquement, tout problème algorithmisable peut devenir un test (ou quiz) utilisable en Moodle.

2. ETAPES A SUIVRE POUR CREER UN TEST PYCLOZE

Les pas à suivre pour créer un test en [PyCloze](#) sont similaires à ceux pour une question calculée (Moodle, Question calculée, 2013). Pour exemplifier la procédure, on considère le problème présenté dans la Figure 1. Les variables (*jokers*) d'entrée sont soulignées en jaune et les variables de sortie sont soulignées en vert.

2.1. Créer une fonction qui résolve le problème

Pour notre exemple, la fonction est :

```
def problem_fun(x):
```

```

mean_age = x['age'].mean()           # calculate the mean age
oldest = len(x['age']) * [""]       # create a list of 3 ""
oldest[np.argmax(x['age'])] = "="   # assign "=" to the oldest child
names_length = sorted(x['name'], key=len)
longest_name = names_length[-1]     # longest name is the last in sorted list
y = {'mean_age': mean_age,
     'oldest': oldest,
     'longest_name': longest_name}
return y

```

2.2. Définir la plage des valeurs pour les variables

Pour les variables jokers, nous voulons 2 x 3 = 6 questions :

- deux listes 'name' contenant trois noms chacune ;
- trois listes 'age' avec les âges correspondant aux trois enfants.

Pour notre exemple, cette étape devient :

```

x_ranges = {'name': np.array(['Antoinette', 'Lawrence', 'Sebastian'],
                             ['Anastasia', 'Catherine', 'Quentin'])),
            'age': np.array([[10.2, 12.9, 9.7],
                             [23.4, 12.5, 17.2],
                             [4.9, 14.1, 8.5]])}

```

2.3. Écrire un texte avec des variables jokers

Pour notre exemple, ce texte est un chaîne de caractères (string) avec les variables jokers (en jaune) et les réponses (en vert) entre des accolades :

```

text = """
John Smith has 3 children:
{name[0]:s}, age {age[0]:.1f},
{name[1]:s}, age {age[1]:.1f},
{name[2]:s}, age {age[2]:.1f}.
The mean age of the children is
$$\mu$$ = {{1:NUMERICAL:={mean_age:.1f}:0.5}} (±0.5).
The eldest child is :
{{1:MULTICHOICE:{oldest[0]:s} {name[0]:s}
~{oldest[1]:s} {name[1]:s}
~{oldest[2]:s} {name[2]:s}}}.
The child with the longest name is {{1:SHORTANSWER:~={longest_name:s}}}.
"""

```

3. CONCLUSIONS

Comme Moodle est accessible avec les navigateurs web les plus populaires depuis un ordinateur, une tablette et/ou un téléphone portable, les tests PyCloze peuvent être utilisés dans :

1. **Les cours magistraux** : des exercices courts (typiquement 5 à 10 minutes) peuvent être proposés pendant les cours pour vérifier la compréhension des notions.
2. **Les travaux dirigés** : une large classe de problèmes proposés dans les TD sont transposable en question à réponses intégrées individualisable par des algorithmes de calcul. PyCloze permet de suivre en temps réel la résolution de problèmes proposés et d'intervenir ponctuellement si un étudiant est en difficulté.

3. **Les examens** ou les **devoirs surveillés** : les examens en présentiel ou en distanciel peuvent se faire en utilisant PyCloze. La correction est pratiquement instantanée.
4. **Les devoirs à la maison** : L'individualisation des problèmes permet la collaboration entre les étudiants sur la méthodologie (ou l'algorithme) de la résolution.

La procédure [PyCloze](#) est sous licence CC-BY et les modules [MoodleCloze.py](#) et [MdlClz.py](#) sont sous licence MIT (Ghiaus, PyCloze v0.1.0, 2021; Ghiaus, PyCloze v0.0.0, 2020). Une banque de questions, [PyCloze-TC](#), pour le *Transfer de chaleur*, est également disponible sous licence CC-BY (Ghiaus, PyCloze-TC v0.0.1, 2020).

REFERENCES

Ghiaus, C. (2020). *PyCloze v0.0.0*. Zenodo: <http://doi.org/10.5281/zenodo.4101668>

Ghiaus, C. (2020). *PyCloze-TC v0.0.1*. Zenodo: <http://doi.org/10.5281/zenodo.4277101>

Ghiaus, C. (2021). *PyCloze v0.1.0*. Zenodo: <http://doi.org/10.5281/zenodo.4655564>

Moodle. (2013). *Question calculée*. Moodle: https://docs.moodle.org/3x/fr/Question_calculée

Moodle. (2020). *Question cloze à réponses intégrées*. Moodle:
https://docs.moodle.org/3x/fr/Question_cloze_à_réponses_intégrées