
Formulaire

Version 1.0.0

Sirine

janv. 06, 2023

1	Introduction	1
1.1	HEADING :	1
1.2	REFERENCES :	1
1.3	Author :	2
2	vivelescollegues_sirine	3
2.1	formulaire_commenté module	3
3	Indices and tables	5
	Index des modules Python	7

This is simple sphinx documentation example.

Here, Explain the project objective and features :

- Ce code a pour but de créer une interface graphique (avec la bibliothèque tkinter) permettant de gérer une liste d'employés. L'interface comporte un formulaire permettant d'ajouter un employé à la liste en entrant son nom, prénom, poste et en cliquant sur le bouton « Enregistrer ». La liste est enregistrée dans un fichier CSV (Comma Separated Values) nommé « liste_employes.csv ».

La fonction « check_form » vérifie que les champs « Prénom » et « Nom » du formulaire ne sont pas vides, et affiche une erreur le cas échéant. Si les champs sont remplis, la fonction appelle la fonction « write_line_in_csv » pour ajouter les données du nouvel employé au fichier CSV, puis affiche un message de validation. La fonction « write_line_in_csv » lit le fichier CSV, ajoute les données du nouvel employé à la fin de celui-ci et réécrit le fichier CSV avec les données mises à jour. L'interface comporte également un listing des employés enregistrés dans le fichier CSV, qui est affiché sous le formulaire.

1.1 HEADING :

- Formulaire vive les colleagues ;

1.2 REFERENCES :

- [Google](#) , search engine has been used throughout the project.
- Other kind of text **Bold** reference.
- **Bold letters**.

1.3 Author :

Sirine Mnassri Yousfi.

2.1 formulaire_commenté module

Module docstring

`formulaire_commenté.check_form(first_name, name, poste)`

Cette fonction vérifie si les champs « Prénom » et « Nom » du formulaire d'ajout d'un employé sont remplis, et affiche un message d'erreur le cas échéant. Si les champs sont remplis, elle ajoute le nouvel employé à la liste des employés et affiche un message de validation.

:first_name [str] Le prénom de l'employé.

:name [str] Le nom de l'employé.

:poste [str] Le poste occupé par l'employé.

None Cette fonction ne retourne rien, mais affiche des messages d'erreur ou de validation et modifie le fichier CSV « liste_employes.csv » si nécessaire.

```
>>> check_form('Alice', '', 'Ingénieur')
# Un message d'erreur est affiché indiquant que le nom est manquant.
>>> check_form('', 'Dupont', 'Ingénieur')
# Un message d'erreur est affiché indiquant que le prénom est manquant.
>>> check_form('Alice', 'Dupont', 'Ingénieur')
# Le nouvel employé Alice Dupont est ajouté à la liste des employés et un message_
↳ de validation est affiché.
```

`formulaire_commenté.write_line_in_csv(first_name, name, date, poste)`

Cette fonction ajoute une nouvelle ligne au fichier CSV contenant la liste des employés. La nouvelle ligne est créée à partir des informations fournies en entrée : nom, prénom, date d'embauche et poste.

:first_name [str] Le prénom de l'employé.

:name [str] Le nom de l'employé.

:date [str] La date d'embauche de l'employé, au format « YYYY-MM-DD ».

:poste [str] Le poste occupé par l'employé.

None Cette fonction ne retourne rien, mais modifie le fichier CSV « liste_employes.csv » en ajoutant une nouvelle ligne.

```
>>> write_line_in_csv('Alice', 'Dupont', '2022-01-01', 'Ingénieur')  
# Le fichier CSV "liste_employes.csv" est modifié et contient maintenant une_  
↪ nouvelle ligne avec les informations sur l'employé Alice Dupont.
```


CHAPITRE 3

Indices and tables

- `genindex`
- `modindex`
- `search`

f

formulaire_commenté, ??