

TP Messenger

UE12 P24 - Python

Objectif

Être capable de s'envoyer des messages via un serveur.

Découpage en étapes

Constat : c'est compliqué -> on y va par étapes

1. On simule le serveur grâce un objet Python
2. On remplacera ensuite l'objet pas à pas

Simuler le serveur

Quelles données sur le serveur ?

- Des utilisateurs
- Des groupes de discussion
- Des messages

On commence par la représentation la plus simple possible :
dictionnaires et listes

Exercice

À partir du fichier `messenger.py` :

1. Ajouter un menu qui permet d'afficher les utilisateurs
2. Ajouter un menu qui permet d'afficher les groupes
3. Ajouter une option pour afficher les messages d'un groupe

Note

À chaque étape :

- Faire un commit
- Refactorer (améliorer l'organisation du code)

Exercice

4. Permettre d'ajouter des utilisateurs

5. Permettre d'ajouter des groupes

Problème : comment sauvegarder les données entre 2
lancements ?

Sauvegarde de données

Plusieurs solutions :

1. Une base de données

- > La bonne solution en temps normal

2. Un fichier

- > Plus simple, donc adapté pour une solution transitoire (en attendant la connexion serveur)

Représenter des données structurées

Plusieurs formats existent :

- CSV
- JSON
- YAML

JSON - Exemple

Un objet : ici un utilisateur

```
1 {  
2   "id": 1,  
3   "name": "My username"  
4 }
```

Une liste : ici des noms

```
1 ["Alice", "Bob", "Charlie"]
```

La combinaison : une liste d'utilisateurs

```
1 [  
2   {"id": 18, "name": "Alice"},  
3   {"id": 5, "name": "Bob"}  
4 ]
```

Note

Les espaces et sauts de ligne ne servent qu'à la lisibilité

Json - lecture en Python

- `json.loads` : pour *parser* une chaîne de caractères
- `json.load` : pour *parser* un fichier

```
1 import json
2
3 my_variable = json.loads('{"id": 35, "name": "Alice"}')
```

Quel sera le type de `my_variable` ?

`dict`

Exercice - Json

1. Déplacer les données du dictionnaire **server** dans un fichier JSON. Vérifier que tout fonctionne.

Note

Si les emojis ne s'affichent pas bien, on verra plus tard pourquoi

2. Ajouter une fonction de sauvegarde du fichier JSON.

Note

Regarder le contenu du fichier json après modification

Modélisation

Les données du serveur sont un peu éparpillées. On peut les regrouper en une **classe**.

Exercice

1. *refactorer* votre code pour créer une classe **Server**
2. Remplacer les autres dictionnaires par des classes



Pensez à bien typer toutes vos données !