

TP2 : gérer le temps

1. Écrivez le programme suivant:

```
1. import time
2.
3. while True:
4.     localtime = time.localtime()
5.     result = time.strftime("%I:%M:%S %p", localtime)
6.     print(result)
7.     time.sleep(1)
```

Enregistrez votre programme dans le fichier `time.py` en allant dans le menu `File/save`. Vous prendrez soin d'enregistrer votre programme dans le dossier `tp2` que vous devez créer.

Exécuter le programme `time.py`, noter le résultat.

2. On modifie le programme précédent pour avoir une exécution à la demande. Pour cela on crée un programme avec un bouton `test`, à chaque fois que l'utilisateur clique sur le bouton `test` le résultat du programme précédent modifié est exécuté (une fois pas de boucle).
3. On s'intéresse à l'automatisation de ce programme en utilisant l'interruption `timer`. La syntaxe de création d'un `timer` est :

threading.Timer(interval, function, args = None, kwargs = None)

Elle permet de Créer un `timer` qui exécutera la fonction « `function` » avec les arguments `args` et `kwargs` après `interval` secondes écoulées. Si `args` est `None` (valeur par défaut), une liste vide sera utilisée.

```
# Program to demonstrate
# timer objects in python
import threading
def horloge():
    # insérer le code précédent
    timer = threading.Timer(5.0, horloge)
    timer.start()
```

rajouter un autre bouton **fin** qui permet de mettre fin au `timer`

```
print("Cancelling timer\n")
timer.cancel()
```

4. On désire mettre en place un feu tricolor (vert , jaune, rouge)

