

# TP - Exercices de Concurrency en Python

Ce TP est composé de plusieurs exercices indépendants pour vous familiariser avec la programmation concurrente en Python, en utilisant des threads et des processus. Chaque exercice doit être réalisé dans le fichier `reponse.py`.

## Exercice 1 : Création de Threads

Créez une fonction `exo1(k: int) -> list[str]` qui crée `k` threads différents. Chaque thread doit ajouter son identifiant (thread ID) à une liste partagée. La fonction doit retourner cette liste après que tous les threads aient terminé leur exécution.

---

## Exercice 2 : Vérification des Nombres Premiers avec Multiprocessing

Créez une fonction `exo2(numbers: list[int]) -> list[bool]` qui prend une liste de nombres et retourne une liste de booléens. Chaque booléen doit indiquer si le nombre correspondant est un nombre premier ou non. Utilisez le module `multiprocessing` pour paralléliser le traitement.

---

## Exercice 3 : Gestion Asynchrone des Tâches

Créez une fonction asynchrone `exo3(tasks: list[WaitAndValidateTask])` qui prend une liste d'instances de `WaitAndValidateTask`. Cette classe doit avoir deux méthodes : - `ask()` qui retourne un temps aléatoire (en secondes) - `validate()` qui doit être appelée après le temps spécifié par `ask()`.

La fonction doit attendre que toutes les méthodes `ask()` soient appelées avant de valider chacune d'elles. Assurez-vous que le timestamp de validation est conforme aux attentes.

---

## Exercice 4 : Création d'un Pool de Processus

Créez votre propre classe `ProcessPool` qui utilise le module `multiprocessing` pour gérer un pool de processus. Cette classe doit pouvoir accepter des tâches via une `Queue`, exécuter ces tâches dans les processus du pool et retourner les résultats (dans une queue dédiée).

Les processus doivent être créés lorsque l'on crée un contexte sur l'instance de la pool. Le contexte est disponible sous forme d'une `Queue` (provenant du module `multiprocessing`)

Le constructeur de `ProcessPool` prend en paramètre uniquement le nombre de processus à créer.

---

## Instructions Générales

- Réalisez chaque exercice dans le fichier `reponse.py`.
- Assurez-vous que tous vos tests passent avant de soumettre votre travail. “