

Formation Python -Programmation objet.

Atelier 5 : Pylint - Tests unitaires.

Exercice : Pylint & Pychecker

Installation de Pylint :

Installez **Pylint** sur votre système (si ce n'est pas déjà fait). Assurez-vous qu'ils sont correctement configurés pour travailler avec Python.

Utilisation de Pylint :

Testez les programmes *test1_pylint.py* et *test2_pylint.py* qui contiennent des erreurs de style et de conventions de codage (par exemple, non-respect de la PEP 8). Utilisez Pylint pour analyser le code et corriger les erreurs détectées.

test1_pylint.py

```
def ma_fonction():
    resultat = 5*5
    print("Le resultat est :" + resultat)
```

test2_pylint.py

```
from random import *

def ma_fonction(n):
    if n<2:
        return \
            False
    fact = 2
    while fact * fact <= n:
        if n % fact == 0:
            return False
        else:
            fact += 1
    return True

for loop in range(100):
    n = randint(1, 100000)
    print(n, ma_fonction(n))
```

Exercice : Mise en Œuvre de Tests Unitaires avec unittest

Tests pour une classe

Créez une classe Python représentant une calculatrice avec des méthodes pour effectuer des opérations mathématiques simples (addition, soustraction, multiplication, division).

Ensuite, écrivez des tests unitaires pour chacune de ces méthodes pour vous assurer qu'elles fonctionnent correctement.

Exercice : Tests pour un gestionnaire de liste :

Créez une classe Python qui implémente un gestionnaire de liste (ajouter un élément, supprimer un élément, vérifier si un élément est présent, etc.). Écrivez des tests unitaires pour cette classe pour vous assurer qu'elle fonctionne correctement dans différentes situations.

Exercice : Cf Atelier 3

A partir du script de l'atelier 3, implémentez les tests unitaires associés