

# Doctest

---

Doctest est un module intégré à la bibliothèque standard de Python qui permet de tester du code Python en utilisant des exemples de session interactifs que l'on trouve souvent dans la documentation. Voici une explication succincte de ce qu'est Doctest :

- **Objectif** : Doctest permet de vérifier que les exemples de code inclus dans la documentation d'un programme Python fonctionnent comme prévu. Il s'agit de tester le code de manière automatisée en utilisant les exemples fournis dans les docstrings.
  - **Fonctionnement** : Pour utiliser Doctest, vous incluez des exemples de code Python dans les docstrings de vos fonctions, classes ou modules. Ces exemples de code ressemblent à des sessions interactives que vous auriez dans l'interpréteur Python. Doctest exécute ces exemples et compare les résultats obtenus avec les résultats attendus, tels qu'ils sont affichés dans les docstrings.
  - **Avantages** : Doctest permet de s'assurer que les exemples de code fournis dans la documentation sont toujours valides et fonctionnent correctement. Cela aide à maintenir la documentation à jour et garantit que les utilisateurs peuvent faire confiance aux exemples fournis pour comprendre et utiliser le code.
  - **Utilisation** : Pour exécuter les tests Doctest, vous pouvez utiliser la ligne de commande en exécutant le module Python avec l'option `-m doctest` ou vous pouvez intégrer les tests Doctest dans votre suite de tests existante en utilisant le module `doctest` dans le cadre de vos tests unitaires. En résumé, Doctest est un outil simple mais puissant pour tester du code Python en utilisant les exemples de session interactifs inclus dans la documentation, ce qui contribue à garantir la qualité et la précision de votre code et de votre documentation.
- 

## Exemples:

Supposons que vous avez une fonction `add` qui prend deux nombres en entrée et renvoie leur somme. Voici à quoi pourrait ressembler la docstring de cette fonction avec des exemples Doctest :

```
def add(a, b):  
    """  
    Cette fonction prend deux nombres en entrée et renvoie leur somme.  
  
    Exemples :  
    >>> add(1, 2)  
    3  
    >>> add(-1, 1)  
    0  
    >>> add(0.5, 0.5)  
    1.0  
    """  
    return a + b
```

Dans cet exemple :

- La docstring de la fonction `add` contient des exemples d'appels à la fonction `add` avec différents arguments.
- Les résultats attendus sont également inclus dans les exemples.
- Lorsque vous exécutez Doctest sur cette fonction, il exécutera automatiquement les exemples de code inclus dans la docstring et vérifiera si les résultats correspondent aux résultats attendus.

Voici comment vous pouvez exécuter Doctest pour cette fonction :

```
if __name__ == "__main__":  
    import doctest  
    doctest.testmod()
```

L'exécution de cette partie de code vérifiera si les exemples inclus dans la docstring de la fonction `add` produisent les résultats attendus. Si tout se passe bien, aucun message ne sera affiché, ce qui signifie que tous les tests sont passés avec succès. Sinon, Doctest affichera des messages d'erreur indiquant les cas où les résultats obtenus ne correspondent pas aux résultats attendus.