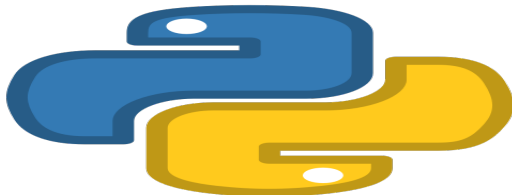


# Python : DocString et Pylint

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



## 1 Documentation

- help
- Documentation HTML
- Documentation et introspection

## 2 Module Pylint

## 3 Extension Flake8

# Python

Considérons la fonction `somme` suivante :

```
def somme(a=0, b=0) :  
    return a + b
```

# Python

## Pour activer l'ajout automatique de commentaire de documentation dans **PyCharm**

- Aller dans `File > Settings` sous **Windows** et **Linux** et `PyCharm > Preferences` sous **Mac** ou tout simplement cliquer sur `Ctrl Alt S`
- Ouvrir `Editor > General > Smart Keys > Python`
- Cocher la case `Insert type placeholders in the documentation comment stub`
- Cliquer sur `Ok`

Pour générer un commentaire de documentation sous **PyCharm**, il faut

- Placer le curseur sur le nom de la fonction `somme`
- Cliquer sur `Alt Enter`
- Cliquer sur `Insert documentation string stub >`

© Achref EL MOUELHI ©

Pour générer un commentaire de documentation sous **PyCharm**, il faut

- Placer le curseur sur le nom de la fonction `somme`
- Cliquer sur `Alt Enter`
- Cliquer sur `Insert documentation string stub >`

Nouveau contenu de la fonction `somme`

```
def somme (a=0, b=0) :  
    """  
  
    :param a:  
    :type a:  
    :param b:  
    :type b:  
    :return:  
    :rtype:  
    """  
    return a + b
```

# Python

## Mettons à jour le commentaire de documentation

```
def somme(a=0, b=0):  
    """calculé et retourne la somme des deux paramètres  
  
    :param a: optionnel, valeur par défaut 0  
    :type a: entier ou réel  
    :param b: optionnel, valeur par défaut 0  
    :type b: entier ou réel  
    :return: résultat de la somme de deux paramètres a et b  
    :rtype: entier ou réel  
    """  
  
    return a + b
```

## Règles à respecter pour la **DocString**

La **PEP 257** définit les conventions de documentation (**DocString**) en **Python**.

- Une documentation pour chaque fonction, commençant par une phrase résumée concise.
- La documentation est à placer immédiatement après la ligne contenant le mot-clé `def`.
- Privilégier les `"""` pour la **DocString**.
- La dernière ligne de la documentation doit seulement contenir les `"""`.



# Python

L'appel de la fonction `help` permet l'extraction du DocString associé

```
help(somme)
# affiche Help on function somme in module __main__:
#
# somme(a=0, b=0)
#     calcule et retourne la somme des deux paramètres
#     :param a: optionnel, valeur par défaut 0
#     :type a: entier ou réel
#     :param b: optionnel, valeur par défaut 0
#     :type b: entier ou réel
#     :return: résultat de la somme de deux paramètres a et b
#     :rtype: entier ou réel
```

# Python

## Plusieurs commandes pour générer la documentation

- `python -m pydoc -b` pour afficher la documentation dans une page **HTML**
- `python -m pydoc -w main` pour générer la documentation pour le fichier `main.py` dans `main.html`
- `python -m pydoc -w .\` pour générer la documentation pour tous les fichiers du projet

# Python

## Quelques constantes sur les fonctions

```
print(somme.__doc__)
# affiche      calcule et retourne la somme des deux paramètres
#      :param a: optionnel, valeur par défaut 0
#      :type a: entier ou réel
#      :param b: optionnel, valeur par défaut 0
#      :type b: entier ou réel
#      :return: résultat de la somme de deux paramètres a et b
#      :rtype: entier ou réel
#

print(somme.__name__)
# somme

print(somme.__defaults__)
# (0, 0)
```

# Python

Pour modifier la DocString de la fonction `somme`

```
somme.__doc__ = "addition"
```

```
print(somme.__doc__)  
# affiche addition
```

# Python

## Pylint

- Outil de vérification de qualité de code écrit en langage **Python**
- Utilisant principalement les propositions de la **PEP 8**
- Documentation : <https://docs.pylint.org/>

© Achret L.

# Python

## Pylint

- Outil de vérification de qualité de code écrit en langage **Python**
- Utilisant principalement les propositions de la **PEP 8**
- Documentation : <https://docs.pylint.org/>

Pour installer `pylint`, exécutez

```
pip install pylint
```

# Python

## Pour vérifier votre code avec **ByCharm**

- Allez dans `Tools > External Tools`
- Cliquez sur `Pylint`

© Achref EL M...

# Python

## Pour vérifier votre code avec **ByCharm**

- Allez dans `Tools > External Tools`
- Cliquez sur `Pylint`

## Ou en exécutant

```
pylint our_file.py
```



# Python

Critères	PyChecker	Pylint
Focus	Détection de bugs	Bugs, style, qualité du code
Conventions ( <b>PEP 8</b> )	Non pris en charge	Pris en charge
Maintenance	Moins actif	Actif
Rapidité	Plus rapide	Plus lent
Supporté par <b>Python 3</b>	Non	Oui

# Python

## Flake8

- Extension **VSC**
- Permettant de vérifier si les propositions de la **PEP 8** sont respectées