

Création de fonctions et de modules

Outils Numériques / Semestre 5
/ Institut d'Optique / B1_2

Utilisation de fonctions

- Intérêts
 - réutilisation d'un même bloc (avec des paramètres différents)
 - meilleure lisibilité du code
 - répartition du travail d'équipe
 - validation simplifiée des fonctionnalités

Définition de fonctions

- Fonction somme

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

```
k = somme( 2.0, 5.1 )  
print( k )
```

7.1

Portée des variables et fonctions

- Variables locales

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

```
print( c )
```

??

```
print(c)
```

NameError: name 'c' is not defined

[https://astrofrog.github.io/py4sci/_static/09.%20Module
s%20and%20Variable%20Scope.html](https://astrofrog.github.io/py4sci/_static/09.%20Module%20and%20Variable%20Scope.html)

Documenter ses fonctions

- Intérêts
 - **réutilisation simplifiée** des fonctions (*quelques jours, mois...*)
 - **redistribution de vos codes** à des tierces personnes (collègues...)

Documenter ses fonctions

- Conventions / ***Python Enhancement Proposal (PEP)***
 - **PEP 8 : guide de style**
 - Recommandations pour aider les développeurs Python à écrire du code compréhensible par des **tierces personnes**
 - **PEP 257 : chaine de documentation**
 - Recommandations pour aider les développeurs à documenter leurs codes et pour générer des documentations au format ***docstring***

PEP 8 / Quelques recommandations

- **noms de variables en minuscules**, avec des underscores (tirets bas). Convention « ***snake_case*** »
- **constantes en majuscules**, avec des underscores
- **noms de classe** avec une majuscule au début de chaque mot, sans ponctuation. Convention « ***CapitalizedCase*** »
- ***Commenter et documenter***
 - phrases complètes en anglais
 - pas de commentaires qui contredisent le code

Documenter ses fonctions

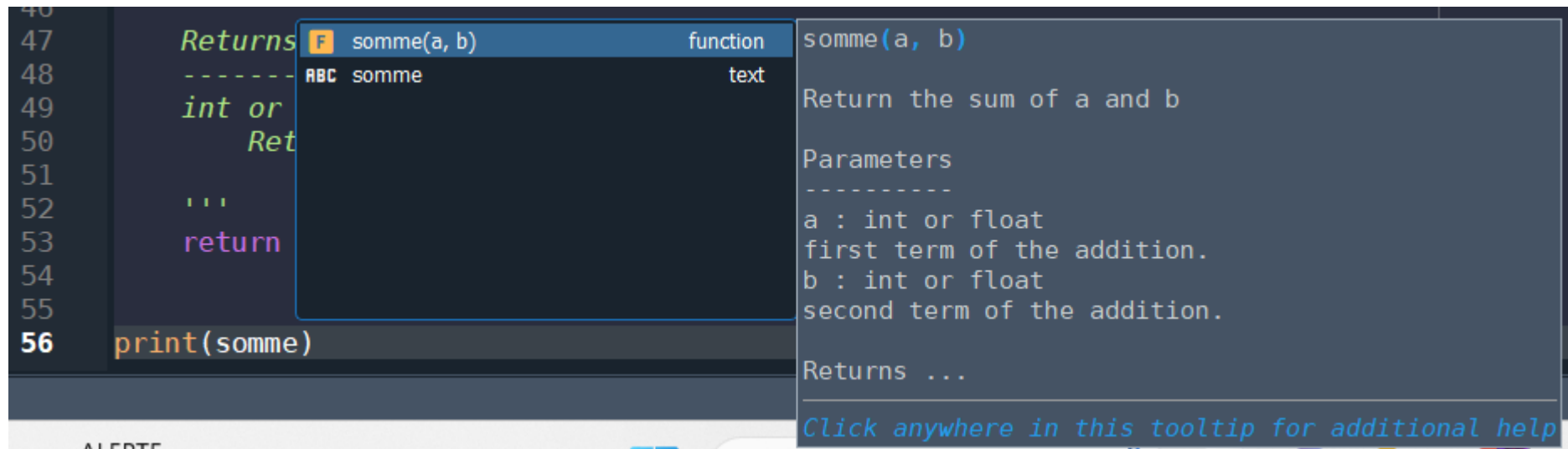
- **Docstring** pour une fonction

```
def somme(a, b):  
    """  
    Return the sum of a and b  
  
    Parameters  
    -----  
    a : int or float  
        first term of the addition.
```

```
    [...]  
  
    b : int or float  
        second term of the addition.  
  
    Returns  
    -----  
    int or float  
        Return the sum of a and b.  
    """  
    return a+b
```


Documenter ses fonctions

- Résultat de la doc



The screenshot shows a code editor with a function definition and a tooltip displaying its documentation. The code in the editor is as follows:

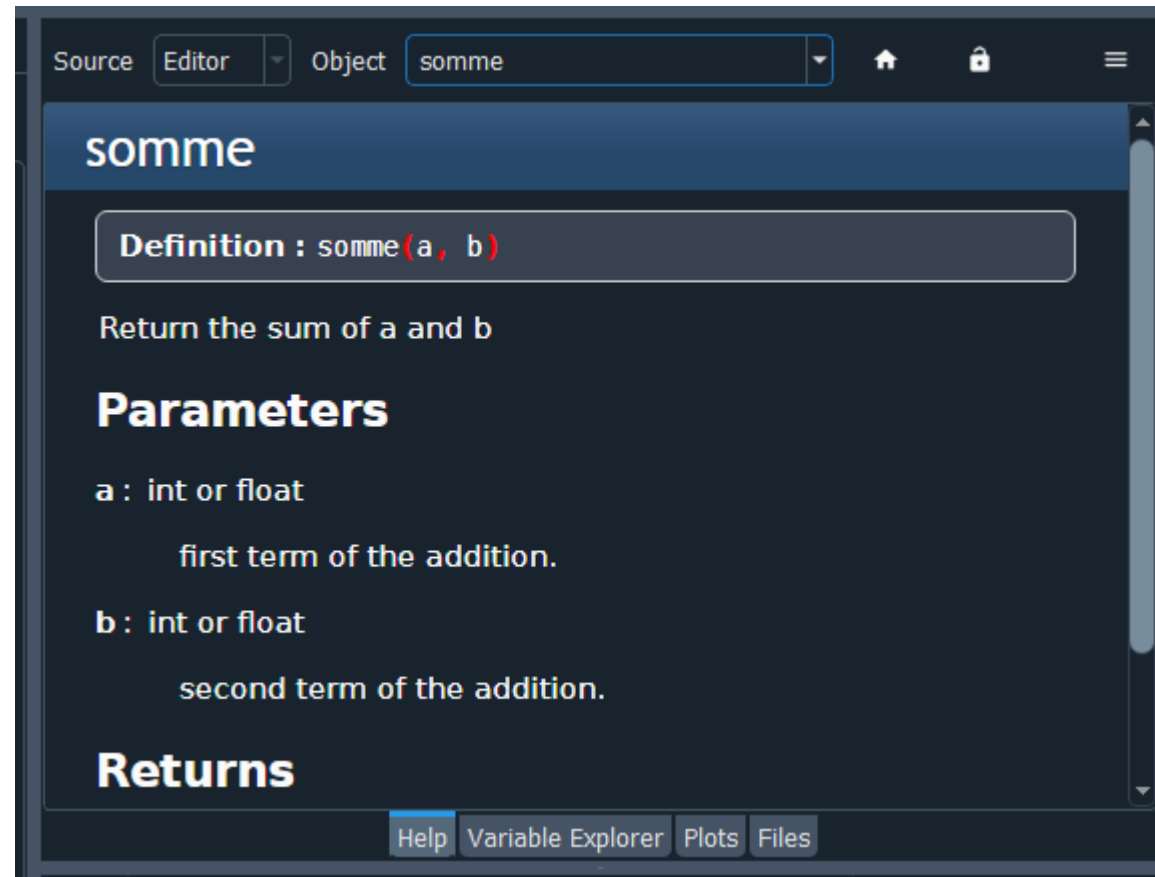
```
40  
47 Returns  
48 -----  
49 int or  
50 Ret  
51  
52 ...  
53 return  
54  
55  
56 print(somme)
```

The tooltip for the function `somme(a, b)` is displayed, showing the following documentation:

```
somme(a, b)  
Return the sum of a and b  
  
Parameters  
-----  
a : int or float  
first term of the addition.  
b : int or float  
second term of the addition.  
  
Returns ...  
  
Click anywhere in this tooltip for additional help
```

Documenter ses fonctions

- Résultat de la doc



Créer des modules

- Regroupement de fonctions dans un fichier .py indépendant
 - Permet de classifier ses fonctions par thématique
 - Permet une réutilisation des fonctions

```
import mymodule  
print( mymodule.somme(2 , 3) )
```

mymodule.py

```
def somme(a, b):  
    [...]
```