

Générer de la documentation python avec *doxygen*

Table des matières

Introduction	2
Doxygen.....	2
Format des commentaires pris en compte par <i>doxygen</i>	2
Syntaxe des commentaires	2
• Documenter un module	3
• Documenter un projet.....	3
• Documenter une fonction	3
Paramétrage de pycharm.....	4
Génération de la documentation.....	5
Procédure générale.....	5
• Le filtre <i>doxypypy</i>	5
• Préparation des ressources	5
• Lancement de doxywizard.....	5
• Paramétrage et génération	5
• Sauvegarde du paramétrage	6
Affichage de la documentation.....	6
Installation et paramétrage des outils sur votre poste personnel	7
Installation de Doxygen	7
Filtre doxypypy pour python.....	7
Fichier de commande <i>py_filter.bat</i>	7
Latex.....	7
Webographie.....	7

Introduction

La documentation d'un projet informatique peut s'avérer fastidieuse à écrire. C'est pourquoi il existe des outils qui extraient directement de l'information du code source pour produire une documentation. En python, les chaînes de caractères qui servent à commenter le programme et par là même à le documenter s'appellent des *docstrings*.

Écrire des *docstrings* offre de nombreux avantages :

- La fonction `help()` affiche cette documentation dans un shell.
- Les outils de programmation tels que les *IDE spyder* ou *pycharm* affichent cette documentation quand le développeur qui utilise votre code, en a besoin.
- On peut générer automatiquement une bonne documentation avec des commandes qui extraient ces *docstrings*.
- C'est un mécanisme standardisé de documentation : tout le monde sait que si c'est là, et que ça a cette forme, c'est une documentation.
- On peut mettre des tests dans les *docstrings*, qui servent alors d'exemples d'utilisation.

A l'instar d'autres outils comme *sphinx* spécialisé pour la documentation *python*, *doxygen* est un logiciel libre de documentation de code possédant des capacités de génération de documentation à partir du code source d'un programme.

Doxygen

Doxygen est un générateur de documentation capable de produire une documentation logicielle à partir du code source d'un programme. Pour cela, il tient compte de la grammaire de la syntaxe du langage dans lequel est écrit le code source, ainsi que des commentaires s'ils sont écrits dans un format adapté.

Doxygen peut traiter des codes sources dans les langages C++, C, Java, Objective-C, Python, PHP, C#...

Il aide à générer une documentation en ligne (html) et/ou un manuel de référence en Latex, pdf, RTF, etc. Il est multi-plateforme et fonctionne sous Linux, MacOSX et Windows. Il est libre et gratuit.

Nous vous expliquons dans la suite de ce tutoriel comment procéder pour lancer doxygen.

Format des commentaires pris en compte par *doxygen*

Syntaxe des commentaires

Nous fournissons ci-dessous quelques éléments de la syntaxe *doxygen* nécessaires pour l'optimisation de la documentation. La documentation complète est disponible ici :

<http://www.doxygen.nl/manual/>

- Documenter un module

Pour documenter un module, on peut utiliser les tags *file*, *author*, *authors*, *date*, *version*...

Pour présenter les commentaires sur plusieurs lignes, on introduit chaque ligne par une * :

```
"""@file nom_module.py description brève du contenu de mon fichier
* Description du contenu du module
* suite de la description du module

@author Christine Fay-Varnier
@date ...
@version ...

"""
```

- Documenter un projet

Pour documenter ensemble d'un projet, dans le script principal ajouter la balise *@mainpage*

```
"""@mainpage descriptif court de mon projet
* suite du descriptif de mon projet
* ...

@author toto, alfred,...
@date ...
@version ...

@file nom_module.py description brève du contenu de mon fichier
* Description du contenu du module
* suite de la description du module

"""
```

- Documenter une fonction

Pour documenter une fonction ou une méthode, il y a plusieurs syntaxes possibles, avec l'IDE *pycharm*, on utilisera celle-ci :

```
def foo(arg1, arg2...):
    """ descriptif court de la fonction sur une ligne. Suite du descriptif
    à la ligne suivante. Etc.

    @param arg1: premier argument
    @param arg2: second argument
    @return: description du résultat retourné
    """
```

On peut aussi utiliser cette syntaxe¹ :

```
def foo(arg1, arg2...):
    """Résumé de la docstring, sur plusieurs lignes

    Contenu détaillé de la docstring.
    Contenu détaillé de la docstring.

    Args:
        arg1 (type): le premier paramètre
        arg2 (type): le second paramètre

    Returns:
        Description du résultat retourné

    Examples:
        Exemples d'utilisation de la fonction
    """
```

Paramétrage de pycharm²

Afin de faciliter la génération de la documentation, on peut paramétrer *pycharm* pour qu'il aide à la génération des docstrings au format *doxygen*.

Pour ce faire :

- Aller dans : *File / Settings / Tools / Python Integrated Tools*
- Dans *Docstring Format* choisir *Epytext*

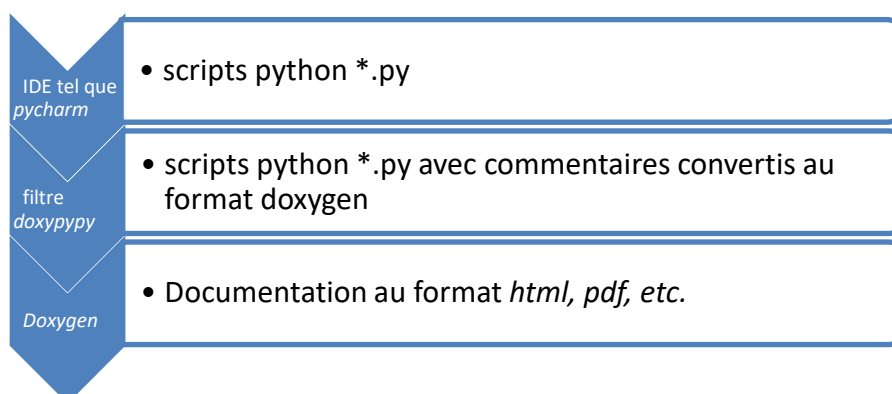
Pour générer la structure d'un docstring d'une méthode :

- Après la saisie de l'entête de cette fonction, taper les `"""` puis taper sur *entrée*. Les lignes à remplir sont automatiquement insérées dans la *docstring*
- Pour ajouter ou modifier *a posteriori* la liste des paramètres de la fonction, cliquer sur le nom de la fonction puis sur l'ampoule qui s'affiche et choisir d'ajouter les paramètres qui manquent.

¹ <https://pypi.python.org/pypi/doxypypy>

² <https://www.jetbrains.com/help/pycharm/settings-tools-python-integrated-tools.html>

Génération de la documentation



Procédure générale

Vous aurez besoin de (voir page 7) Installation et paramétrage des outils sur votre poste personnel:

doxygen <https://doxygen.nl/download.html>

le module *doxypy* <https://pypi.org/project/doxypy/>

py_filter.bat (à télécharger sur Arche)

et si vous voulez générer du pdf : *LaTeX*

- **Le filtre *doxypy***

Doxygen a un support limité pour Python. Il reconnaît les commentaires Python, mais considère le langage comme étant plus ou moins similaire à *Java*. Le module *doxypy* permet d'intégrer des commandes *Doxygen* dans des docstrings Python et de les convertir à la volée en commentaires *Doxygen*.

Un fichier de commande *py_filter.bat* doit exister dans un dossier de programme Windows (sous le dossier *doxygen* ou *Anaconda3* par exemple). S'il n'existe pas, il faut le créer, la procédure pour sa création est décrite en fin de document (voir *Fichier de commande py_filter.bat* page 7).

- **Préparation des ressources**

On conseille de séparer les scripts de la documentation :

- Regrouper les scripts à documenter dans un dossier particulier, par exemple *Scripts*
- Créer un dossier pour la génération de la documentation, par exemple *Documentation*

- **Lancement de *doxywizard***

L'interface *doxywizard* associée à *Doxygen* permet de paramétrer la génération de la documentation

- lancer *doxywizard*

- **Paramétrage et génération**

- Identification du projet**

- fixer les paramètres concernant votre projet dans l'onglet *Wizard*
 - dossier dans lequel se trouve le projet à documenter
 - dossier dans lequel trouver les scripts documentés sous le dossier du projet (par exemple *Scripts*)

- dossier dans lequel générer la documentation (par exemple *Documentation*)

Paramètres généraux

- dans le volet *Wizard / Mode*, cocher l'option *Optimize for Java or C# output*
- dans le volet *Wizard / Output* cocher *plain HTML* ou bien *with navigation panel*
- dans le volet *Expert / Project* dans *OUTPUT_LANGUAGE* préciser la langue de génération
- dans le volet *Expert / Build* cocher l'option *EXTRACT_ALL*
- dans le volet *Expert / Input* ajustez les paramètres :
 - dans *FILTER_PATTERNS* ajouter : **.py=py_filter*
Cela permet d'exécuter le fichier de commande *=py_filter.bat*

Génération de la documentation

- dans l'onglet *Run* lancer *Run doxygen*
 - On peut afficher le résultat de la génération en cliquant sur *Show HTML output* dans l'onglet *Run*
- **Sauvegarde du paramétrage**

On peut enregistrer les paramètres courants pour pouvoir les réutiliser à l'aide de la commande *File / Save*

Affichage de la documentation

La documentation est générée dans le dossier spécifié dans les paramètres, par exemple dans le dossier *Documentation*.

- La documentation sous format *HTML* est disponible dans le dossier *html*. Il suffit de cliquer sur le fichier *index.html* pour l'afficher dans votre navigateur.
- Pour générer une documentation en format *pdf*, il est nécessaire de disposer d'un compilateur *Latex*. Si c'est le cas, la documentation en format *pdf* est générée en cliquant sur le fichier de commande *make.bat*. Le fichier généré s'appelle *refman.pdf*

Installation et paramétrage des outils sur votre poste personnel

Installation de Doxygen

- télécharger *doxygen* : <https://doxygen.nl/download.html>
- installer *doxygen*

Filtre doxypypy pour python

Le module *doxypy* permet de convertir les *docstrings* python en commentaires *doxygen*.

Pour l'installer sur votre ordinateur, le plus simple est d'utiliser la commande *pip* :

- ouvrir une invite de commande (*cmd.exe*)
- taper dans cette ligne la commande : *pip install doxypypy*
- et tout se fait tout seul

Si la commande *pip* n'est pas reconnue, taper la suite de commandes suivantes dans la fenêtre de commande :

- *cd /*
- *cd Anaconda3*
- *cd Scripts*
- *pip install doxypypy*

Fichier de commande *py_filter.bat*

- sous windows, avec un éditeur, *blocnotes*, *pycharm*, *notepad*, créer un fichier de commande contenant :

```
python -m doxypypy.doxypypy -a -c %1
```

- enregistrez ce fichier sous le nom *py_filter.bat* dans le dossier *doxygen* ou *Anaconda3*
- il faut absolument l'enregistrer dans un dossier de commandes connu dans le *PATH* de *windows*

Latex

- Si vous ne disposez pas de Latex sur votre poste personnel, il suffit de copier le dossier *latex* de la documentation sur un des postes de l'école et de lancer le fichier de commandes *make.bat* depuis ce poste.
- Si vous souhaitez installer Latex, nous vous conseillons d'installer *MiKTeX*

Webographie

<https://www.doxygen.nl/index.html>

<https://cyberzoide.developpez.com/tutoriels/java/doxygen/>