



GESTION DE LA DOCUMENTATION



DIFFÉRENTS SCÉNARIOS

- Specifications du code
- Documentation du code
- Manuels / Aide utilisateurs
- HowTo / Tutoriels

AVANT DE COMMENCER VOTRE DÉVELOPPEMENT LOGICIEL

- Une part importante dans la vie d'un logiciel se déroule en dehors du code.
- Cela commence souvent par l'écriture des pré-requis et des spécifications, qui seront ensuite convertis en code.

PENDANT LE DÉVELOPPEMENT

Quand doit-on écrire la documentation de son code ?

PENDANT LE DÉVELOPPEMENT

Quand doit-on écrire la documentation de son code ?

1. Principalement avant
2. Parfois pendant
3. Rarement après

PENDANT LE DÉVELOPPEMENT

Que doit-on écrire dans son code ?

PENDANT LE DÉVELOPPEMENT

Que doit-on écrire dans son code ?

- Tout ce qui pourrait ne pas être évident pour quelqu'un d'autre que l'auteur (y compris l'auteur dans un an).

PENDANT LE DÉVELOPPEMENT

Que doit-on écrire dans son code ?

- Tout ce qui pourrait ne pas être évident pour quelqu'un d'autre que l'auteur (y compris l'auteur dans un an).
- En pratique:
 - pre et post-conditions pour les méthodes
 - utilisation prévue des variables
 - tout ce qui est susceptible d'être mal compris par quelqu'un qui n'aurait pas écrit l'intégralité du code.

LE CODE EST AUSSI DE LA DOCUMENTATION

```
def toto(a, b):  
    """ worst case scenario  
    """  
    return b[a]
```

```
def extract_value_from_dict(key: str, data: dict) -> float:  
    """ better scenario  
    """  
    return data[key]
```

DANS LE CODE

Automatiser la génération de documentation : Doctest

```
def function(arg):  
    """  
    A short description.  
    A bit longer description.  
  
    Parameters  
    -----  
    arg : type  
        description  
  
    Returns  
    -----  
    type  
        description  
  
    Raises  
    -----  
    Exception  
        description  
  
    Examples  
    -----  
    Examples should be written in doctest format, and  
    should illustrate how to use the function/class.  
    >>>
```

AUTOUR DU CODE

Une fois le code écrit, est-ce que la documentation va au-delà ?

AUTOUR DU CODE

Une fois le code écrit, est-ce que la documentation va au-delà ?

- Souvent oui: écriture d'un manuel utilisateur, tutoriels, publication de la documentation en ligne ou en ligne de commande

AUTOUR DU CODE

Une fois le code écrit, est-ce que la documentation va au-delà ?

- Souvent oui: écriture d'un manuel utilisateur, tutoriels, publication de la documentation en ligne ou en ligne de commande
- Cela va dépendre de la portée de votre travail. Identifiez les utilisateurs.

AUTOUR DU CODE

PUBLICATION DE LA DOCUMENTATION EN LIGNE

- Il existe des outils pour facilement extraire la documentation du code, et la formater pour une publication sur le web
 - Exemple : [Sphinx](#)
- Ces outils permettent aussi d'étoffer la documentation, et d'ajouter du contenu autour du code.
- L'hébergement peut se faire facilement sous GitLab/GitHub (avec l'intégration continue).
- Voir aussi [readthedocs](#) ou [mkdocs](#)

LE TUTORIEL

L'ARME DE CONVERSION MASSIVE

- Mieux que 20,000 lignes d'explication, mieux qu'un long discours, l'exemple est un outil puissant pour présenter le fonctionnement du code.
- Un bon tutoriel bénéficie du retour de la communauté d'utilisateurs

☰ QUELQUES BONNES PRATIQUES

- La documentation doit être un processus continu, comme les tests.
- Suivez le style de code correspondant à votre langage.
- Envisagez de créer un template au sein de votre équipe pour la documentation au niveau du projet.
- Utiliser des outils pour générer de la documentation à partir du code source.
- Utiliser les outils de l'IDE.
- Identifier les utilisateurs du code pour cibler au mieux la documentation.
- Utiliser des outils, et automatiser le plus possible la chaîne!