Documentation en Python

La documentation est un moyen de communication clé de la programmation, car elle rend votre code plus lisible, facilite la collaboration et permet aux autres développeurs de comprendre et d'utiliser votre code.

La documentation en Python est essentielle pour expliquer le but, le fonctionnement et l'utilisation des modules, des fonctions, des classes et des variables.

Il existe plusieurs types de documentation couramment utilisés pour rendre le code plus compréhensible et accessible.

Elle est également utile pour générer automatiquement des documentations conviviales pour les utilisateurs de vos bibliothèques et modules.

Commentaires

Les commentaires sont le moyen le plus simple de documenter votre code. Ils sont destinés aux développeurs qui lisent le code source et ne sont pas inclus dans la documentation générée automatiquement.

```
# Ceci est un commentaire simple
```

Docstrings

Les docstrings sont des chaînes de caractères incluses dans le code source qui décrivent une fonction, une classe, un module ou une variable. Ils sont utilisés pour générer une documentation automatique.

```
def addition(a, b):
    """
    Cette fonction renvoie la somme de deux nombres.

    Args:
        a (int): Le premier nombre.
        b (int): Le deuxième nombre.

    Returns:
        int: La somme des deux nombres.
    """
    return a + b
```

Documentation de Module

La documentation de module est généralement placée au début d'un fichier de code et décrit le contenu, le but et l'utilisation du module.

```
"""
Ce module contient des fonctions pour effectuer des opérations mathématiques de
base.
"""

def addition(a, b):
    return a + b
```

Documentation de Classe

La documentation de classe est utilisée pour décrire les classes et leurs méthodes.

```
class Personne:
    """
    Cette classe représente une personne avec un nom et un âge.
    """

    def __init__(self, nom, age):
        """
        Initialise une nouvelle instance de la classe Personne.

        Args:
            nom (str): Le nom de la personne.
            age (int): L'âge de la personne.
        """
        self.nom = nom
        self.age = age
```

Documentation de Fonction

La documentation de fonction est utilisée pour décrire les fonctions et leurs paramètres. Elle est généralement placée sous la signature de la fonction.

```
def division(dividende, diviseur):
    """
    Cette fonction renvoie le résultat de la division.

    Args:
        dividende (float): Le nombre à diviser.
        diviseur (float): Le nombre par lequel diviser.

    Returns:
        float: Le résultat de la division.

    Raises:
        ZeroDivisionError: Si le diviseur est égal à zéro.
    """
    if diviseur == 0:
        raise ZeroDivisionError("Impossible de diviser par zéro.")
    return dividende / diviseur
```

Documentation de Variable

La documentation de variable est utilisée pour expliquer le but d'une variable, en particulier lorsqu'elle n'est pas évidente.

```
nombre_de_clients: int = 100  # Le nombre total de clients dans la base de données
```

Annotations de Type

Bien que ce ne soit pas une forme de documentation explicite, l'annotation de type est également un moyen de documenter le code en indiquant le type attendu des paramètres et des valeurs de retour des fonctions.

```
def addition(a: int, b: int) -> int:
    return a + b
```

Structure de documentation de fichier

Les `${variable}` représentent les informations à changer.

Il est possible de faire un fichier de configuration `.env` à la racine du projet pour chaque personne qui travaille sur le projet et d'importer les informations.

Exemple de fichier qui récupère les informations issues du .env

```
"""
Titre : Documentation fichier Python
Nom du projet : ${PROJECT_NAME}

Nom du fichier : ${NAME}

Date de la dernière révision : ${MONTH_NAME_FULL},${YEAR},${MONTH},${DAY},${TIME}

Auteur(s) : Prenom NOM
Auteur(s) : ${USER}

IDE utilisé : ${PRODUCT_NAME}

Client : e-learning

#-----------------------------------------------

name='$Package_name',
    version='$Version',
    packages=$PackageList,$PackageDirs
    url='$URL',
    license='$License',
    author='$Author',
    author_email='$Author_Email',
    description='$Description'
```

```
#-------------------------------------------------

Fichiers du projet :
                    -   Lab1_4.py
"""

# Indispensable hors contexte de PyCharm->en bas à droite
# -*- coding:Utf-8 -*-

#################################################
#               Programme principal
#################################################

#-----------------------------------------------
#           Zone des 'imports' de modules
#-----------------------------------------------

import pandas as pd

#-------------------------------------------------------
#       Zone de déclaration des variables 'globales'
#-------------------------------------------------------

ma_variable = 4

#-------------------------------------------------------------
#       Zone de déclaration des modules ou des fonctions
#-------------------------------------------------------------

def calcul...

#-------------------------------------------------------------
#                       PROGRAMME
#-------------------------------------------------------------
```