04 - Kommentare und Dokumentation

Zur Dokumentation des Sourcecodes verwenden wir Docstring. Dadurch ersparen wir uns eine separate Beschreibung der Klassen und Methoden in einem Textdokument.

Klassen

Der Docstring einer Klasse ist die Visitenkarte einer Klasse. Er informiert den Programmierer über Aufgabe und Version dieser Klasse.

	Jede Klasse hat Beschreibung als Docstring. Dieser enthält mindestens die Angaben zu:
	* Kurzbeschreibung
	* Attribute
	* Methoden (ausser Standard-Getter, -Setter, -Deleter)

Beispiel

```
class Category:
    """
    the category to which a project is assigned

    Attributes
    ----------
    category_id : int
        unique key for a category
    title : str
        the title of a category

    Methods
    -------
    compare(self, other):
        compares the title of this category with another category
    """
```

Funktion

Der Docstring liefert alle Angaben über die Aufgabe und Schnittstelle einer Funktion.

	Jede Methode hat einen Docstring mit den Angaben:· Kurzbeschreibung· Beschreibung der Parameter (falls vorhanden)· Beschreibung des Returnwerts (falls vorhanden)· Beschreibung der Exceptions (falls vorhanden)

Beispiel

```
def load_project(project_id, category)
    """
    loads a project from the database

        :param: project_id (int): the unique id of a project
        :param: category (str):   the category title to be searched

        :return: result_code(str): a resultcode (SUCCESS, NOT FOUND, ERROR)
    """
```

Kommentare

Kommentare sind wie das Salz in der Suppe: Zuwenig und es schmeckt nicht, zu viel und es ist ungeniessbar.

	Jeder Programmblock, dessen Aufgabe nicht offensichtlich ist, wird in einem Kommentar beschrieben.
	Ein kurzer Blockkommentar ist oftmals sinnvoller als Zeilenkommentare, die über den Bildschirm hinausreichen.

Beispiel: Blockkommentar

```
# create a new connector, if not exists
# and establish connection if needed


if self._connector == None:
    self._connector = Connector
if self._connector.get_handle == None:
    self._connector.connect
```

Beispiel: Zeilenkommentar

Kurze Kommentare können auch hinter den Befehl geschrieben werden.

```
self._school_class = None  # this reference follows later in time
self._name         = name
self._report       = report # immediately establish the two-way relationship
here
report.set_student(self)
```

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/howto/codingstandards/04-kommentare

Last update: **2024/08/13 06:34**