

## Programmieren in Python

### 13. Docstrings und weitere Kleinigkeiten

Robert Mattmüller

Albert-Ludwigs-Universität Freiburg

Handlungsplanungs-Praktikum  
Wintersemester 2010/2011

1 / 11

## Docstrings

- ▶ Dokumentation des Programms durch Kommentare im Code.
- ▶ Blockkommentare ähnlich zu Javadoc- oder Doxygen-Kommentaren.
- ▶ Docstring: Stringliteral am Anfang
  - ▶ eines Moduls,
  - ▶ einer Funktion,
  - ▶ einer Klasse oder
  - ▶ einer Methode.
- ▶ Wichtig für: Module, öffentliche Methoden.

2 / 11

## Docstrings: Beispiel

docstrings\_demo.py

```
"""
This string will become the module docstring of the
docstrings_demo module when the file is imported.
"""

class MyClass(object):
    """The class's docstring"""

    def my_method(self):
        """The method's docstring"""
        do_something()

def my_function():
    """The function's docstring"""
    do_something_else()
```

3 / 11

## Docstrings: Verwendung

- ▶ Notation: In dreifachen Anführungszeichen.
- ▶ Position: Anfang des Moduls bzw. Anfang des Funktions-, Methoden- oder Klassenkörpers.
- ▶ Formatierung Einzeiler:
  - ▶ Öffnende Anführungszeichen, eigentlicher Text und schließende Anführungszeichen in einer Zeile.
  - ▶ Keine Leerzeilen vor oder nach Docstring.
  - ▶ Ganzer Satz (englisch), etwa „Do this.“ oder „Return that.“
- ▶ Formatierung Mehrzeiler:
  - ▶ Erst Zusammenfassingszeile, dann Leerzeile, dann ausführlichere Beschreibung.
  - ▶ Einheitliche Einrückung.
  - ▶ Bei Kommandozeilenskripten: Usage-String.

4 / 11

## Docstrings: Zugriff auf Text

Docstrings können auf unterschiedliche Arten gelesen/genutzt werden. Sei `documented` die dokumentierte Klasse, Funktion, Methode bzw. das dokumentierte Modul, auf das sich der Docstring bezieht.

- ▶ Über spezielles Attribut `documented.__doc__`
- ▶ Über Aufruf von `help(documented)`
- ▶ Über mit externem Werkzeug generierte Dokumentation:
  - ▶ Werkzeuge: `pydoc`, `epydoc`, ...
  - ▶ Ausgaben: HTML, PDF, Graphviz, ...

5 / 11

## Docstrings: Beispiele für `__doc__`

### Python-Interpreter

```
>>> import docstrings_demo
>>> print(docstrings_demo.__doc__)
```

This string will become the module docstring of the `docstrings_demo` module when the file is imported.

```
>>> print(docstrings_demo.MyClass.__doc__)
The class's docstring
>>> print(docstrings_demo.MyClass.my_method.__doc__)
The method's docstring
>>> print(docstrings_demo.my_function.__doc__)
The function's docstring
```

Funktioniert auch für eingebaute Module, Klassen etc.

6 / 11

## Docstrings: Beispiele für `help`

### Python-Interpreter

```
>>> import docstrings_demo
>>> help(docstrings_demo)
>>> help(docstrings_demo.MyClass)
>>> help(docstrings_demo.MyClass.my_method)
>>> help(docstrings_demo.my_function)
```

Funktioniert auch für eingebaute Module, Klassen etc.

7 / 11

## Docstrings: Beispiele für `help(docstrings_demo)`

```
Help on module docstrings_demo:
NAME
  docstrings_demo
FILE
  /home/robert/pythonkurs/docstrings_demo.py
DESCRIPTION
  This string will become the module docstring of the
  docstrings_demo module when the file is imported.
CLASSES
  builtins.object
    MyClass
class MyClass(builtins.object)
  | The class's docstring
  |
  | Methods defined here:
  |
  | my_method(self)
  |     The method's docstring
  |
  | ...
```

8 / 11

Aufruf:

Shell

```
epydoc [--html|--pdf] [-o DIR] [-v|-q] OBJECTS...
```

Unterstützt einfaches Markup („Epytext“). Details dazu unter

- ▶ <http://epydoc.sourceforge.net/manual-epytext.html> und
- ▶ <http://epydoc.sourceforge.net/manual-fields.html>.

Hier nur ein kleines Beispiel.

square\_with\_docstrings.py

```
def square(x):  
    """  
    Return the square of M{x}. The B{square} of a number  
    is the number times itself.  
  
    @type x: number  
    @param x: The number to be squared.  
    @rtype: number  
    @return: The square of M{x}, i.e., M{x*x}.  
    """  
    return x*x
```

- ▶ M{x} setzt x im mathematischen Modus.
- ▶ B{x} druckt x fett.
- ▶ @type, @param, @rtype, @return: hoffentlich selbsterklärend.

## Weitere Kleinigkeiten

Wir haben in den bisherigen Lektionen mindestens zwei wichtige Sprachkonstrukte bzw. Methoden aus Zeitgründen nicht behandelt und werden sie in diesem Kurs auch nicht mehr behandeln. Das sind:

- ▶ Die with-Anweisung.
- ▶ Die Methode format der Klasse string.

Da beide nicht ganz unwichtig sind, hier zwei Verweise, wo ihr weiterlesen könnt:

- ▶ Für with-Anweisung:  
<http://effbot.org/zone/python-with-statement.htm>
- ▶ Für str.format()-Methode:  
<http://docs.python.org/py3k/library/string.html#formatstrings>