

Python Tutorial: Doctest

Doctest ist ein Modul in Python, das es ermöglicht, Tests in den Docstrings eines Programms zu schreiben. Diese Tests können dann automatisch ausgeführt werden, um zu überprüfen, ob der Code wie erwartet funktioniert. Doctest ist besonders nützlich für einfache Tests und um Beispiele in der Dokumentation zu verifizieren.

Ähnliche Tests gibt es beispielsweise in [Rust](#), [Haskell](#) und [Scala](#).

Einführung in Doctest

Das Doctest-Modul sucht nach interaktiven Python-Sessions in den Docstrings eines Programms und führt diese als Tests aus. Dies bedeutet, dass die Beispiele in Ihrer Dokumentation gleichzeitig als Tests dienen können, was hilft, die Dokumentation aktuell zu halten und sicherzustellen, dass die Beispiele wie erwartet funktionieren.

Grundlagen von Doctest

Einen einfachen Test schreiben

Um Doctest zu verwenden, schreiben Sie einfach Beispiele für interaktive Python-Sessions in die Docstrings Ihrer Funktionen, Klassen oder Module. Ein Beispiel könnte so aussehen:

```
def add(a, b):  
    """  
    Addiert zwei Zahlen und gibt das Ergebnis zurück.  
  
    >>> add(2, 3)  
    5  
    >>> add(-1, 1)  
    0  
    """  
    return a + b
```

Doctests ausführen

Um die Doctests auszuführen, können Sie das Doctest-Modul als Skript ausführen und dabei die Datei angeben, die getestet werden soll. Alternativ können Sie Doctest in Ihrem Code mit der `doctest.testmod()`-Funktion aufrufen.

Ausführung von der Kommandozeile

```
python -m doctest -v deine_datei.py
```

Ausführung im Code

```
import doctest
doctest.testmod() # Prüft alle Doctests im aktuellen Modul
```

Was Doctest überprüft

Doctest vergleicht die tatsächliche Ausgabe der interaktiven Session mit der erwarteten Ausgabe, wie sie im Docstring angegeben ist. Es berücksichtigt dabei nicht nur den Wert, sondern auch die genaue Formatierung der Ausgabe.

Vorteile und Einschränkungen

Vorteile

- **Einfachheit:** Doctest ist leicht zu lernen und zu verwenden, besonders für einfache Tests und Dokumentationsbeispiele.
- **Dokumentation:** Fördert gut dokumentierten Code, indem Beispiele direkt in die Dokumentation integriert und getestet werden.

Einschränkungen

- **Komplexität:** Für komplexere Tests oder solche, die spezielle Setup- oder Teardown-Schritte erfordern, sind andere Testframeworks wie `unittest` oder `pytest` besser geeignet.
- **Formatierung:** Doctest ist sehr genau in Bezug auf die Ausgabeformatierung, was manchmal zu Frustration führen kann, wenn Tests aus scheinbar trivialen Gründen fehlschlagen.

Zusammenfassung

Doctest ist ein nützliches Werkzeug in Python, um einfache Tests direkt in den Docstrings zu schreiben und auszuführen. Es eignet sich hervorragend, um Beispiele in der Dokumentation zu verifizieren und sicherzustellen, dass diese Beispiele korrekt bleiben, während sich der Code weiterentwickelt.