

# LU09.A00: PyTests finden



Erstelle ein JSON-Array mit allen Unittests in einem Python-Projekt.

## Aufgabe

Das Skript durchsucht ein Projekt nach Dateien, die Unittests enthalten. Die Dateinamen müssen `test_` oder `_test` enthalten. Zum Beispiel:

- `test_main.py`
- `main_test.py`

Innerhalb dieser Dateien sucht das Skript nach Funktionen, deren Bezeichner mit `test_` beginnt. Die Bezeichner dieser Funktionen werden in eine Liste geschrieben. Schlussendlich wird diese Liste sortiert und als JSON-Array ausgegeben.

## Beispiel

```
[  
  "test_figure_init_empty",  
  "test_figure_add_shape",  
  "test_figure_add_shape_multiple",  
  "test_figure_take_shape",  
  "test_figure_remove_shape_by_title",  
  "test_figure_remove_shape_by_index",  
  "test_figure_total_area_scale1",  
  "test_figure_total_area_scale3",  
  "test_figureshape_init_empty",  
  "test_figureshape_set_figure",  
  "test_figureshape_set_shape",  
  "test_main"  
]
```

## Wieso Python und Bash?

Grundsätzlich könnte man die Aufgabe ausschliesslich mit Python lösen oder als reines Bash-Skript schreiben. Jede Sprache hat jedoch ihre Vorteile, die wir hier ausnutzen wollen:

- Dateien durchsuchen: Hier spielt Bash seine Stärke aus
- Arbeiten mit Collections: Dies lässt sich leichter in Python umsetzen.

## Umsetzung

In den folgenden Aufgaben werden Sie zunächst einzelne Teilschritte umsetzen und testen. Zum Schluss schreiben Sie die `main`-Funktion um alle Teilfunktionen zu verbinden.

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



Permanent link:  
[https://wiki.bzz.ch/modul/m122/learningunits/lu09/aufgaben/find\\_tests?rev=1734338242](https://wiki.bzz.ch/modul/m122/learningunits/lu09/aufgaben/find_tests?rev=1734338242)

Last update: **2024/12/16 09:37**