

LU04.A18 - Slicing



Verwende Python's Slicing-Mechanismus, um bestimmte Teile einer Liste oder eines Strings zu extrahieren und zu manipulieren. Dabei sollen alle Spezialfälle abgedeckt werden.

Aufgabenstellung

Erstelle eine Python-Funktion namens `slicing_special_cases`, die folgende Aufgaben erledigt:

1. Extrahiere die ersten drei Elemente der Liste `[1, 2, 3, 4, 5]`. Die Ausgabe sollte `[1, 2, 3]` sein.
2. Extrahiere die letzten zwei Elemente der Liste `[1, 2, 3, 4, 5]`. Die Ausgabe sollte `[4, 5]` sein.
3. Kehre die Liste `[1, 2, 3, 4, 5]` um. Die Ausgabe sollte `[5, 4, 3, 2, 1]` sein.
4. Extrahiere jeden zweiten Wert der Liste `[1, 2, 3, 4, 5]`. Die Ausgabe sollte `[1, 3, 5]` sein.
5. Extrahiere die Elemente zwischen den Indizes `-4` und `-1` der Liste `[1, 2, 3, 4, 5]`. Die Ausgabe sollte `[2, 3, 4]` sein.
6. Extrahiere die ersten drei Elemente der Liste `[1, 2, 3, 4, 5]` und kehre sie um. Die Ausgabe sollte `[3, 2, 1]` sein.
7. Extrahiere die letzten zwei Elemente der Liste `[1, 2, 3, 4, 5]` und kehre sie um. Die Ausgabe sollte `[5, 4]` sein.
8. Kehre die Liste `[1, 2, 3, 4, 5]` um und nimm nur jeden zweiten Wert. Die Ausgabe sollte `[5, 3, 1]` sein.
9. Extrahiere die Elemente zwischen den Indizes `-4` und `-1` der Liste `[1, 2, 3, 4, 5]`, kehre sie um und nimm nur jeden zweiten Wert. Die Ausgabe sollte `[4, 2]` sein.

Code Vorlage

[main.py](#)

```
def slicing_special_cases():
    """Üben von Slicings in Python."""

    # Liste für alle Übungen
    lst = [1, 2, 3, 4, 5]

    # Extrahiere die ersten drei Elemente der Liste [1, 2, 3, 4, 5].
    first_three = ...

    # Extrahiere die letzten zwei Elemente der Liste [1, 2, 3, 4, 5].
    last_two = ...
```

```
# Kehre die Liste [1, 2, 3, 4, 5] um.
reversed_lst = ...

# Extrahiere jeden zweiten Wert der Liste [1, 2, 3, 4, 5].
every_second = ...

# Extrahiere die Elemente zwischen den Indizes -4 und -1 der Liste
[1, 2, 3, 4, 5].
between_neg4_and_neg1 = ...

# Extrahiere die ersten drei Elemente der Liste [1, 2, 3, 4, 5] und
kehre sie um.
first_three_reversed = ...

# Extrahiere die letzten zwei Elemente der Liste [1, 2, 3, 4, 5]
und kehre sie um.
last_two_reversed = ...

# Kehre die Liste [1, 2, 3, 4, 5] um und nimm nur jeden zweiten
Wert.
reversed_every_second = ...

# Extrahiere die Elemente zwischen den Indizes -4 und -1 der Liste
[1, 2, 3, 4, 5], kehre sie um und nimm nur jeden zweiten Wert.
between_neg4_and_neg1_reversed_every_second = ...

return {
    'first_three': first_three,
    'last_two': last_two,
    'reversed_lst': reversed_lst,
    'every_second': every_second,
    'between_neg4_and_neg1': between_neg4_and_neg1,
    'first_three_reversed': first_three_reversed,
    'last_two_reversed': last_two_reversed,
    'reversed_every_second': reversed_every_second,
    'between_neg4_and_neg1_reversed_every_second':
between_neg4_and_neg1_reversed_every_second
}

if __name__ == '__main__':
    print(slicing_special_cases())
```

Vorgehen

1. Akzeptiere das GitHub Classroom Assignment
2. Klone dein persönliches Repository in die Entwicklungsumgebung

3. Implementiere die Funktion `slicing_special_cases`

Abgabe

Die Abgabe der Lösung erfolgt als Push in das persönliche GitHub-Repository.

⇒ *GitHub Repo für externe Besucher*

GitHub Repository <https://github.com/templates-python/m323-lu04-a18-slicing>

Lernende am BZZ **müssen** den Link zum GitHub Classroom Assignment verwenden



Kevin Maurizi

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

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu04/aufgaben/slicing>

Last update: **2024/03/28 14:07**

