Slicing in Python

Einführung in das Slicing

- Slicing ist eine Technik in Python, um Teile einer Sequenz (z.B. Listen, Strings, Tupeln) auszuwählen.
- Es ermöglicht den Zugriff auf Teilbereiche von Datenstrukturen ohne die Daten zu verändern.
- Besonders nützlich für Datenmanipulation und -analyse.

Slicing in Python

Slicing-Syntax

Die grundlegende Syntax für Slicing ist: [start:stop:step]

- start: Der Index, an dem das Slicing beginnt (inklusive).

  Optional. Standardwert: 0, wenn nicht angegeben.

- stop: Der Index, an dem das Slicing endet (exklusive).

  Optional. Standardwert: Das Ende der Sequenz, wenn nicht angegeben.

- step: Die Schrittweite zwischen den Elementen.

  Optional. Standardwert: 1, wenn nicht angegeben.

Beispiel

```
my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(my_list[1:8])  # [1, 2, 3, 4, 5, 6, 7]
print(my_list[1:8:3])  # [1, 4, 7]
```

Slicing in Python

Erweiterte Slicing-Techniken

my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Negative Schritte:

my_list[4:1:-1] wählt Elemente in umgekehrter Reihenfolge.

```
print(my_list[4:1:-1])  #[4, 3, 2]
```

Umkehren von Sequenzen:

[::-1] kehrt die Liste um.

```
print(my_list[::-1])  #[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

Kopie der gesamten Sequenz erstellen:

[:] wird verwendet, um eine Kopie der gesamten Sequenz zu erstellen.

```
print(my_list[:])   #[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Slicing in Python

Erweiterte Slicing-Techniken

my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Einfügen von Elementen mit Slicing:

```
new_elements = [100, 101]
my_list[5:5] = new_elements
print(my_list)  # Ausgabe: [0, 1, 2, 3, 4, 100, 101, 5, 6, 7, 8, 9]
# my_list[5:5] spezifiziert die Position direkt nach dem Element mit Index 4 (ab index 5,also nach der  Zahl 4),
# vor dem Element mit Index 5 (also vor der Zahl 5).
```

Ersetzen von Elementen mit Slicing:

```
my_list[5:7] = [200, 201, 202]
print(my_list) # Ausgabe: [0, 1, 2, 3, 4, 200, 201, 202, 7, 8, 9]
# my_list[5:7] spezifiziert den Bereich von Index 5 bis 6 und ersetzt die Elemente in diesem Bereich.
```

Slicing in Python

Der slice-Objekt

Das slice-Objekt erlaubt es, Slicing-Parameter dynamisch zu erstellen.

- Syntax: slice(start, stop, step)

```
#Die Verwendung der Parameter bei slice(start, stop, step) ist identisch mit der Nutzung der Parameter in der Slicing-Syntax [:].

my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

my_slice = slice(1, 4)

print(my_list[my_slice])
```

- Vergleich zu [:]: slice kann als Variable gespeichert und wiederverwendet werden.

- Nützlich für dynamische Slicing-Operationen.

Slicing in Python

Zusammenfassung

- Slicing ist eine wichtige Technik, um mit Sequenzen in Python zu arbeiten.
- [:] ist die gebräuchlichste Syntax für Slicing, während slice-Objekte mehr Flexibilität bieten.
- Durch das Verständnis von Slicing kann man effizienter und klarer mit Datenstrukturen arbeiten.

Mehr erkunden:

https://www.w3schools.com/python/python_strings_slicing.asp

https://www.geeksforgeeks.org/string-slicing-in-python/

https://www.simplilearn.com/tutorials/python-tutorial/python-slicing