Der Datentyp Tupel

Ein unveränderlicher, sequentieller Datentyp

Der Tupel

Man kann sich ein Tupel als eine unveränderliche Liste vorstellen. Sprich, wenn es einmal erstellt wurde, kann sein Inhalt nicht mehr verändert werden.

Das bedeutet, dass weder Elemente hinzugefügt noch Elemente gelöscht werden können.

Tupel werden auf die gleiche Art definiert wie Listen, nur dass statt der eckigen Klammern nun runde Klammern verwendet werden.

Vorteile von Tupeln

Tupel sind in seltenen Fällen performanter als Listen.

Wenn bekannt ist, dass Daten nicht geändert werden müssen, sollten Tupel Listen vorgezogen werden, um ungewollte Änderungen zu vermeiden.

Der wichtigste Vorteil liegt aber darin, dass Tupel auch als Schlüssel in Dictionaries verwendet werden können, da Schlüssel nur Objekte von unveränderlichen Datentypen sein können.

Listen können dazu nicht verwendet werden.

Elemente eines Tupels über Index adressieren

x = 1, 2

y = (2, 3,)

z = 3,

p = y[0]

Ein Tupel mit tuple erstellen

Mit der Funktion tuple lässt sich aus einer Sequenz ein Tupel erstellen

```
a = [1, 2, 3, 4]
z = tuple(a)
print(z)
=> (1, 2, 3, 4)

a = "1234511ab"
z = tuple(a)
print(z)
=> ('1', '2', '3', '4', '5', '1', '1','a','b')
```

Tupel als Rückgabewert von Funktionen

Der Tupel bildet eine elegante Möglichkeit, mehrere Werte aus einer Funktion zurückzugeben.

```
def fn():
    return 2, 3


x, x = fn()
```

Tuple Unpacking

Tuples und Sequenzen im Allgemeinen können entpackt werden.

```
x, y, z = (1, 2, 3)
x, y, *z = (1, 2, 3, 4, 5)
```