



Strings

Inhalt

- [Strings](#)
 - [String-Methoden](#)

In der letzten Einheit wurden String-Variablen schon kurz vorgestellt. Diese dienen dazu, Zeichenketten aufzunehmen und wiederzugeben. Strings werden in Anführungszeichen (" oder ') gestellt. In den bisherigen Beispielen kamen bereits zwei Operatoren zum Einsatz, um String-Variablen zu manipulieren:

Das Stern-Symbol (*). Dieses kommt bei Zahlen für die Multiplikation zum Einsatz. Du kannst jedoch auch eine Zeichenkette mit einer Zahl "multiplizieren", in diesem Fall erhältst du bei der Ausgabe eine Wiederholung der entsprechenden Zeichen in der vorgegebenen Anzahl.

Außerdem haben wir bereits das Pluszeichen (+) verwendet. Dieses hat, neben der mathematischen Addition von Zahlen, zwei Zeichenketten miteinander verbunden. In den folgenden Abschnitten lernst du weitere Befehle kennen, mit denen es möglich ist, String-Variablen zu bearbeiten.

Wichtig ist es zunächst, auf einen einzelnen Teil der Zeichenkette zugreifen zu können. Das ist möglich, indem du hinter den Variablennamen in eine eckige Klammer ([] bei Windows mit Alt Gr + 8 und Alt Gr + 9, bei macOS mit alt + 5 und alt + 6) die Position des entsprechenden Buchstabens schreibst. Dabei musst du lediglich berücksichtigen, dass der erste Buchstabe den Index 0 hat. Daher muss die entsprechende Zahl immer um 1 kleiner sein, als die eigentliche Position.

Willst du beispielsweise auf den Buchstaben auf Position 4 zugreifen, musst du diesen Buchstaben über den Index 3 ansteuern.

	W	i	l	l	k	o	m	m	e	n
Index	0	1	2	3	4	5	6	7	8	9

Häufig ist es nicht nur notwendig, einen einzelnen Buchstaben aus der Zeichenkette herauszulösen, sondern einen größeren Bereich. Auch das ist möglich. Hierfür musst du ebenfalls eine eckige Klammer hinter den Variablennamen schreiben. Darin stehen der Start- und der Endpunkt mit einem Doppelpunkt voneinander getrennt. Dabei musst du



berücksichtigen, dass das Zeichen, das du als Endpunkt angibst, nicht mehr ausgegeben wird, sondern nur alle Zeichen, die vor dieser Position stehen. Wenn du am Anfang der Zeichenkette beginnen willst, dann kannst du dafür den Index 0 eingeben. Alternativ dazu ist es möglich, die Zahl vor dem Doppelpunkt wegzulassen. Dann beginnt das Programm automatisch am Anfang der Zeichenkette. Wenn du die Zahl nach dem Doppelpunkt weglässt, beginnt das Programm am angegebenen Startpunkt und gibt alle Zeichen bis zum Ende der Zeichenkette wieder.

Das folgende Beispielprogramm zeigt dir, wie du Teile aus einer String-Variablen herauslösen kannst:

```
variable = "Herzlich Willkommen!"

print(variable)
print(variable[0])
print(variable[5])
print(variable[3:7])
print(variable[:8])
print(variable[9:])
print(variable[-1])
```

Wenn du dieses Programm ausführst, siehst du genau, wie der Zugriff auf einzelne Teile der Zeichenkette erfolgt.

Um das Beispielprogramm übersichtlicher zu gestalten, könntest du jeweils noch einen String einfügen, der dir in der Ausgabe anzeigt, welchen Index du angesteuert hast. So ist die Ausgabe am Ende einfacher zu verstehen:

```
variable = "Herzlich Willkommen!"

print(variable)
print("[0]")
print(variable[0])
print("[5]")
print(variable[5])
print("[3:7]")
print(variable[3:7])
print("[:8]")
print(variable[:8])
print("[9:]")
print(variable[9:])
print("[-1]")
print(variable[-1])
```

Interessant ist in dem Beispielprogramm der letzte Befehl `print(variable[-1])`. Man



könnte meinen, dass es keine negativen Positionen geben kann. Das Minuszeichen steht jedoch dafür, dass das Programm beim Zählen der Position von hinten beginnt. Die Position -1 bezeichnet daher das letzte Zeichen, die Position -2 das vorletzte und so weiter.

	W	i	l	l	k	o	m	m	e	n
Index	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

Eigentlich wäre es logisch, für das letzte Zeichen die Position -0 zu verwenden. Das ist mathematisch jedoch gleichbedeutend mit 0, sodass der Interpreter das erste Zeichen ausgibt. Daher musst du immer mit -1 beginnen, wenn du beim letzten Zeichen beginnen willst.

String-Methoden

Neben diesen einfachen Operationen mit Zeichenketten gibt es noch viele weitere, "schlauere" Möglichkeiten der Bearbeitung: Es handelt sich um Funktionen, die in Python integriert sind und die sich durch einen ganz bestimmten Schlüsselbegriff abrufen lassen. Diese Funktionen werden als String-Methoden bezeichnet. Die Vielfalt ist dabei so groß, dass es in diesem Workbook nicht sinnvoll wäre, jede einzelne Funktion zu beschreiben. Die wichtigsten sollen jedoch kurz vorgestellt werden.

Um diese Methoden anzuwenden, ist es in der Regel notwendig, die Variable, die die Zeichenkette enthält, dem entsprechenden Schlüsselbegriff durch einen Punkt getrennt voranzustellen. Danach steht eine Klammer, die jedoch in vielen Fällen leer bleiben kann. Manchmal, je nach Methode, ist es allerdings notwendig, hier einen bestimmten Parameter einzufügen, damit die Methode richtig funktioniert. Um sie auszuprobieren, soll ein kleines Programm mit verschiedenen Methoden als Beispiel dienen:

```
text = "herzlich willkommen!"

print(text.capitalize())
print(text.count("i"))
print(text.endswith("en!"))
print(text.find("e"))
print(text.isdigit())
print(text.upper())
print(text.replace("e", "0"))
```

Führe das Programm durch einen Klick auf Run aus und du erhältst folgendes Ergebnis in der Konsole:



```
C:\Users\PC\PycharmProjects\Erstes_Programm>liste.py
Herzlich willkommen!
2
True
1
False
HERZLICH WILLKOMMEN!
h0rzlich willkomm0n!

C:\Users\PC\PycharmProjects\Erstes_Programm>
```

Die Ausgabe des Programms

Die Methode `count()` benötigt als Parameter eine weitere Zeichenkette innerhalb der Klammer. Sie gibt an, wie häufig die in der Klammer angegebene Zeichenkette in der ursprünglichen Variable vertreten ist. Damit lässt sich beispielsweise die Häufigkeit eines Buchstabens oder eines Wortteils bestimmen. In längeren Texten kannst du damit auch bestimmte Wörter zählen.

Die Methode `endswith()` benötigt ebenfalls eine Zeichenkette in der Klammer und gibt eine boolesche Variable zurück. Diese ist `True`, wenn die ursprüngliche Variable genau mit den angegebenen Zeichen endet. Tut sie das nicht, ist der Wert `False`.

Die Methode `find()` sucht nach einzelnen Zeichen oder Zeichenketten innerhalb der Variable und gibt deren Position zurück. Sollten sie mehrfach auftreten, erhältst du den Wert, an dem sie das erste Mal erscheint. Sollte das Zeichen überhaupt nicht vorkommen, erhältst du den Wert `-1` zurück.

Die Methode `isdigit()` überprüft, ob eine Zeichenkette ausschließlich aus Ziffern besteht. Je nach Ergebnis der Überprüfung gibt sie `True` oder `False` zurück.

Die Methode `upper()` verwandelt alle Kleinbuchstaben in der Zeichenkette in Großbuchstaben. Bei Ziffern, Satzzeichen oder bei Großbuchstaben hat sie keinen Effekt.

Mit der `replace()`-Methode kannst du einen Buchstaben oder eine bestimmte Abfolge durch einen anderen Ausdruck ersetzen. Diese Methode benötigt zwei Parameter. Der erste von ihnen enthält die Buchstabenfolge, die du ersetzen möchtest. Der zweite gibt den neuen Wert an.

Hierbei handelt es sich nur um einen kleinen Ausschnitt aus den vielfältigen Methoden, die du auf Zeichenketten anwenden kannst. Eine vollständige Übersicht der String-Methoden



findest du in der offiziellen [Python-Referenz](#).

Probiere an dieser Stelle selbst ein paar Methoden mit deinem eigenen Beispieltext aus, du kannst nichts kaputt machen!

An dieser Stelle kannst du dir schonmal das Projekt [GUI](#) anschauen. Zwar empfehlen wir, zunächst das dafür empfohlene Vorwissen zu sammeln – du siehst aber schonmal, was auf dich zukommt, wenn du später deine Daten nicht immer nur in der Konsole eingeben bzw. ausgeben willst, sondern in einer grafischen Benutzeroberfläche.

Prüfe deine Skills: [Aufgabe 3](#)