

Empfohlene Unterrichtsinhalte als Vorbereitung für dieses Thema: 01-02 (Was ist Python?), 01-03 (Variablen), 02-01 (Strings), 02-02 (Listen)

Einführung in die computerlinguistische Programmierung mit Python

03-01: Indexing und Slicing

Sowohl Strings als auch Listen sind Sequenzen, die aus einzelnen Elementen bestehen. Mit dem Befehl `len` können wir prüfen, wie viele Elemente in so einem Objekt enthalten sind:

In []:

```
# Wir definieren die Variable "fish" als Liste.
# Die Liste hat drei Elemente, die mit Kommas
# voneinander getrennt sind.
fish = ["cod", "halibut", "whale"]
print(fish)
print(len(fish))

# Wir definieren die Variable "whale" als String.
# Wie viele Elemente hat der String?
whale = "Moby Dick"
print(whale)
print(len(whale))
```

In Listen oder Strings hat jedes Element seinen eigenen Platz. Die Reihenfolge spielt bei diesen Datentypen immer eine Rolle!

In []:

```
print("Seyffarth" == "Seyffahrt")    # :(
print("Seyffarth" == "Seyffrath")   # :( :(
print("Seyffarth" == "Seyfarth")    # :( :( :(
```

Wir können mithilfe eines **Index** herausfinden, was für ein Element an einer bestimmten Stelle einer Sequenz steht. Dazu beginnen wir links beim ersten Element mit `0` und zählen nach rechts weiter. Den Index, der uns interessiert, schreiben wir in eckige Klammern nach der Sequenz:

In []:

```
# Welches Zeichen steht an Index 5?
print("kurzer String"[5])
```

Bei Listen sieht das etwas gewöhnungsbedürftig aus:

In []:

```
# Welche Zahl wird ausgegeben?
print([1,2,3][1])
```

In []:

```
# Meistens arbeiten wir aber sowieso mit Variablen.  
# Dann sieht das Indexing eher so aus:  
fish = ["magicarp", "horsea", "corphish"]  
print(fish[2])
```

Wenn man vom Ende der Sequenz zählen möchte, verwendet man negative Zahlen: `seq[-1]` ist der letzte Index einer Sequenz namens `seq`, `seq[-2]` der vorletzte usw. Jedes Element der Sequenz kann also mit einem positiven Index oder mit einem negativen Index ausgewählt werden.

In []:

```
# Welche Buchstaben werden hier jeweils ausgegeben?  
# Achtung! Wir zählen links bei 0 los,  
# von rechts fangen wir mit -1 an!  
  
print("ABCDE"[0])  
print("ABCDE"[-1])  
print("ABCDE"[3])  
print("ABCDE"[-2])
```

Mit **Slicing** können wobei wir Teilsequenzen aus der ursprünglichen Sequenz extrahieren. Wir geben dazu den Startindex und den Endindex an, in der Form `seq[start:end]`.

In []:

```
print("ABCDE"[0:1])  
print("ABCDE"[0:3])  
print("ABCDE"[0:0]) # Leer :(  
print("ABCDE"[-1:-3]) # Leer :(  
print("ABCDE"[-3:-1])
```

Der Startindex ist dabei **Teil der gewählten Sequenz**, während der Endindex **außerhalb** der Teilsequenz liegt. Beim Verwenden negativer Indizes müssen wir genau wie bei positiven Indizes die kleinere Zahl zuerst angeben (siehe Beispiel oben).

Wenn die Teilsequenz am Anfang der ursprünglichen Sequenz beginnen oder am Ende der Sequenz enden soll, können wir diesen Index einfach weglassen.

In []:

```
print("ABCDE"[:3])  
print("ABCDE"[2:])  
print("ABCDE"[:])
```

In []:

```
# Was wird hier ausgegeben? Und warum?  
print("ABCDE"[-4:3])
```

Weder Indexing noch Slicing verändert die Originaldaten! Wir können also unbesorgt nacheinander auf verschiedene Indizes oder Slices zugreifen.

In []:

```
fish = ["cod", "halibut", "cow"]
print(fish)
print("-----")

# Welchen Datentyp haben die verschiedenen Slices?
print(fish[1])
print(fish[-3:1])
print(fish[1:])

print("-----")

# Hurra, die Variable enthält immer noch alle Elemente!
print(fish)
```

Wenn eine Sequenz Elemente enthält, deren Datentyp ebenfalls eine Sequenz darstellt, können wir Indexing oder Slicing auch verschachteln:

In []:

```
fish = ["cod", "halibut", "cow"]
print(fish)
print(fish[0])

# Welches Element wird hier ausgewählt?
print(fish[0][-1])
```

Zusammenfassung

- Strings und Listen sind **Sequenzen** von Elementen.
- Die Reihenfolge der Elemente spielt bei Listen und Strings eine Rolle.
- Wir können den Index einer Position angeben und so herausfinden, welches Element an dieser Stelle der Sequenz steht.
- Wir können einen Start- und einen Endindex angeben und so herausfinden, welche Teilsequenz zwischen den angegebenen **Indizes** steht.
- Indexing und Slicing können verschachtelt werden, falls die Datentypen das erlauben.

Weitere Themen dieser Woche

- 03-02: Bedingungen
- 03-03: Schleifen
- 03-04: Zufallszahlen