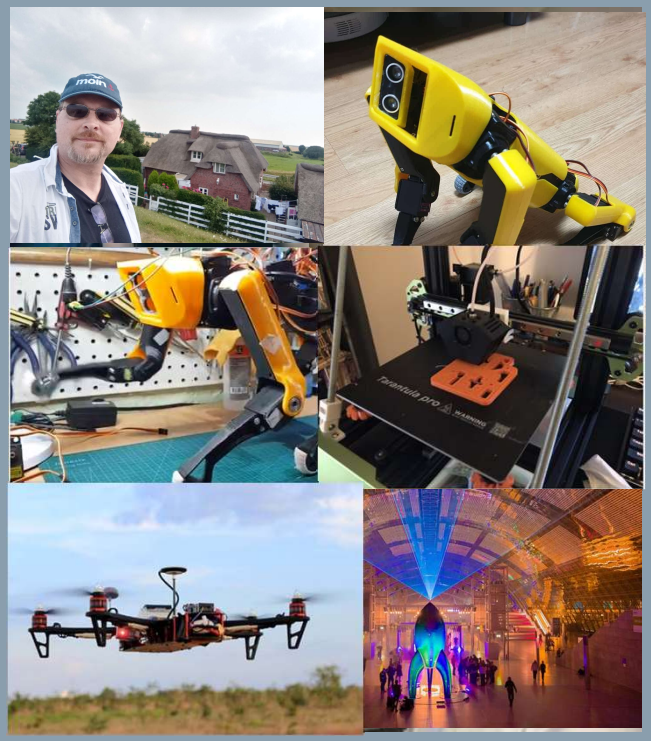
Cloud Command

Python Programmieren

Cloud Command

Zum Referenten:



Andreas Schmidt

- Jahrgang 1975, wohnhaft in Hamburg
- Fachinformatiker für Systemintegration
- Android-App Entwickler
- Java Entwickler
- Administrator für Heterogene Netzwerke
- Kommunikationselektroniker
- Über 20 Jahre im IT-Support und Consulting tätig
- ITIL
- Hardware-Entwicklung

Certified by EXIN
EXIN Ethical Hacking Foundation

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Im letzten Kapitel haben wir Listen kennengelernt und deren grundsätzliche Möglichkeiten.
- Da geht aber noch deutlich mehr und sehr einfach.
- Wir haben eine Liste erstellt und komplett ausgegeben über:

```
vornamen = ['Axel', 'Elke', 'Martin']
print(vornamen)
```

- Eine Liste erweitert über:

```
vornamen = ['Axel', 'Elke', 'Martin']
vornamen += ['Heike', 'Sabine']
print(vornamen)
```

- Und ein einzelnes Element der Liste ausgegeben.
- Wichtig war, dass der Index einer Liste immer bei 0 beginnt!

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Im Folgenden lernen wir alle wichtigen Möglichkeiten der Listen-Methoden kennen:
  - Element in Liste am Ende einfügen
  - Element an bestimmter Position in Liste einfügen
  - Element vom Listenende löschen
  - Element von bestimmter Position der Liste löschen
  - Element anhand vom Wert aus einer Liste löschen

- Aber was sind Methoden?

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Methoden und Listen

- Bisher sind wir im Python Kurs noch nicht direkt auf Methoden zu sprechen gekommen.
- Aber was sind diese Methoden, die es in jeder Programmiersprache gibt?
  - Sucht man nach Synonymen von dem Wort „Methode“ stößt man auf „Vorgehen, Arbeitsweise und Handhabung“.
  - Wir können also auf ein bestimmtes Objekt (in unserem Fall sind die Objekte der Begierde die Listen) vorgegebene Methoden (sprich Möglichkeiten) anwenden.
- Wer sich schon ein bisschen mit objektorientierter Programmierung auseinandergesetzt hat, kennt das.
  - Wir haben ein Objekt und das Objekt hat einerseits Inhalt (sprich Attribute) und mögliche vordefinierte Funktionen (sprich Methoden).

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Und hier kommt auch die Schreibweise zum Vorschein.
- Als Erstes wird das Objekt genannt und dann wird die Methode an dieses Objekt mit einem Punkt „verkettet“.
- Gegebenenfalls kann man noch weitere Parameter für die Funktionsweise der Methode mit übergeben.
- Klarer wird es im konkreten Beispiel (natürlich hier mit Listen).

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Nutzen wir die Liste mit den Vornamen und lassen uns den zweiten Namen (sprich Indexnummer 1) ausgeben:

```
vornamen = ['Axel', 'Elke', 'Martin']
print(vornamen[1])
```

- Jetzt wenden wir die Methode upper() auf unser Objekt (in diesem Fall einen String) an.
- Unsere Ausgabe vom Listenobjekt mit dem Index 1 wird mit der Methode „schreib alles groß“ bzw. upper() verbunden:

```
vornamen = ['Axel', 'Elke', 'Martin']
print(vornamen[1].upper())
```

- Als Ergebnis bekommen wir:

```
ELKE
```

- Der Inhalt der Liste selber ändert sich nicht, sondern nur die Ausgabe wird in Großschreibung gebracht.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Hier haben wir das erste Beispiel für eine Methode, die keine weiteren Angaben braucht.
  - Es soll ja einfach alles in Großbuchstaben geschrieben werden.

- Schauen wir im nächsten Beispiel Methoden an, die weitere Angaben benötigen, damit sie ihre Funktion auch erfüllen können.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Weiterer Eintrag am Ende einer Liste anhängen: append()

- Wir wollen in einer Liste einen weiteren Eintrag anhängen.
- Dazu gibt es eine Methode mit dem Namen append().
- Hier ist sofort einsichtig, dass wir etwas übergeben müssen, was dann an die bestehende Liste angehängt werden wird.
- Unsere „Anhäng-Methode“ wird also immer mit zusätzlichem Parameter in Erscheinung treten:

  append('anzuhängende Eintrag')

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Wir erstellen eine leere Liste mit dem Namen „buchstaben“.

```
buchstaben = []
print(buchstaben)
```

- Wir haben also mit der Liste „buchstaben“ ein Objekt.
- Diesem Listenobjekt wollen wir einen weiteren (in diesem Fall den ersten) Eintrag anhängen.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Wir haben also mit der Liste „buchstaben“ ein Objekt.
- Diesem Listenobjekt wollen wir einen weiteren (in diesem Fall den ersten) Eintrag anhängen.
- Also wenden wir append() an:

```
buchstaben = []
print(buchstaben)
buchstaben.append('a')
buchstaben.append('b')
print(buchstaben)
```

- Als Ausgabe erhalten wir:

```
[]
['a', 'b']
```

- In der ersten Zeile sehen wir, dass eine leere Liste existiert.
- Dann wird über append() zweimal ein Buchstabe angefügt.
- Dieser wird jeweils am Ende der Liste eingetragen!

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Element in Liste an bestimmte Position einfügen: insert()

- Wollen wir nicht am Ende, sondern an einer vordefinierten Stelle Inhalt in der Liste einfügen, kommt die Methode insert() zum Einsatz.

- Weil es so schön übersichtlich ist, nutzen wir wieder unser Buchstabenbeispiel und wollen nun in der bestehenden Liste
  - mit ['a', 'b'] ein „c“ zwischen „a“ und „b“ schieben (sprich einfügen).

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Jetzt benötigen wir bei der Methode insert() bereits 2 Parameter:
    - was soll eingefügt werden
    - wo soll es einfügt werden
- Beide Angaben werden in den Runden Klammern der Methode mit übergeben.
- Vergisst man Angaben, erhält man den Fehler
    - „TypeError: insert() takes exactly 2 arguments (1 given)“.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

- Die Reihenfolge ist:
  - wo soll es eingefügt werden als Index
  - was soll eingefügt werden

```
buchstaben = ['a', 'b']
print(buchstaben)
buchstaben.insert(1, 'c')
print(buchstaben)
```

- Das Programm ergibt dann folgende Ausgabe:

```
['a', 'b']
['a', 'c', 'b‘]
```

.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Element aus Liste entfernen: del

- Genauso wichtig wie einfügen ist auch das Löschen.
- In vielen Spielen werden Daten in Listen gespeichert.
- Sind die Gegner in Listen gespeichert und fällt ein Gegner einem heimtückischen Anschlag zum Opfer und verstirbt tragisch, muss dieser aus der Liste entfernt werden.
- Dazu bietet Python die Anweisung del, der ein Index mit übergeben werden muss, welches Listenelement gelöscht werden soll.
- Unser Buchstabenbeispiel:

```
buchstaben = ['a', 'b']
print(buchstaben)
del buchstaben[0]
print(buchstaben)
```

- Und als Ergebnis kommt:

```
['a', 'b']
['b']
```

- Hier wurde mal eine Anweisung (und keine Methode eingeschmuggelt)!
- Aber es gibt auch Methoden, um Elemente aus Listen zu löschen.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Letztes Element aus Liste entfernen: pop()

- Der Vorteil unserer Methode pop() ist,
  - dass wir als Rückgabewert den Inhalt des gelöschten letzten Listeneintrags bekommen
  - und damit weiterarbeiten können (wenn wir das wollen).
- Wichtig ist, pop() löscht immer den letzten Eintrag!

```
buchstaben = ['a', 'b']
print(buchstaben)
letztereintrag = buchstaben.pop()
print(buchstaben)
print("Letzter gelöschter Eintrag: ", letztereintrag)
```

- Und als Ausgabe kommt:

```
['a', 'b']
['a']
```

- Letzter gelöschter Eintrag: b

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Elemente aus Liste entfernen anhand seines Wertes: remove()

- Wir können auch ein Element anhand seines Wertes aus einer Liste entfernen lassen. Dazu wird der Wert mit übergeben:

```
buchstaben = ['a', 'b', 'c']
print(buchstaben)
buchstaben.remove('b')
print(buchstaben)
```

- Ergibt als Ergebnis:

```
['a', 'b', 'c']
['a', 'c']
```

- Versuchen wir einen Wert zu löschen, der nicht in der Liste existiert, erhalten wir die Fehlermeldung „buchstaben.remove('e')
- ValueError: list.remove(x): x not in list”
- Diese Fehlermeldungen können wir später über bestimmte Techniken abfangen, jetzt müssen wir vorerst damit noch leben.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Sortieren von Listen über sorted()

- Benötige man die Listeneinträge sortiert, kann die über die Funktion sorted() durchgeführt werden.
- Im folgenden Beispiel liegen die Buchstaben in der Liste in einer nicht definierten Reihenfolge vor.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Im Folgenden Beispiel erfolgt die erste Sortierung aufsteigend und dann im nächsten Schritt absteigend:

```
buchstaben = ['a', 'c', 'b']
print(buchstaben)
buchstaben_sortiert = sorted(buchstaben)
print(buchstaben_sortiert)
buchstaben_sortiert_absteigend = sorted(buchstaben, reverse=True)
print(buchstaben_sortiert_absteigend)
```

- Das Ergebnis:

```
['a', 'c', 'b'] # unsortierte Liste
['a', 'b', 'c'] # sortiert aufsteigend
['c', 'b', 'a'] # sortiert absteigend
```

.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Soll bei der Sortierreihenfolge nicht auf Groß- und Kleinschreibung geachtet werden, muss die entsprechende Anweisung mitgegeben werden.

```
buchstaben = ['a', 'c', 'B', 'A']
print(buchstaben)
# sortiert, Groß-Kleinschreibung wird beachtet
buchstaben_sortiert_1 = sorted(buchstaben)
print(buchstaben_sortiert_1)
# sortiert ohne Rücksicht auf Großbuchstaben
buchstaben_sortiert_2 = sorted(buchstaben, key=str.lower)
print(buchstaben_sortiert_2)
```

Wir erhalten nun wie erwartet als Ausgabe:

```
['a', 'c', 'B', 'A'] # unsortierte Liste
['A', 'B', 'a', 'c'] # sortiert, Groß-Kleinschreibung wird beachtet
['a', 'A', 'B', 'c'] # sortiert ohne Rücksicht auf Großbuchstaben
```

.

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Alle wichtige Möglichkeiten für Listen

Somit haben wir alle wichtigen Möglichkeiten der Listen-Methoden kennengelernt:

- Element in Liste am Ende einfügen
- Element an bestimmter Position in Liste einfügen
- Element vom Listenende löschen
- Element von bestimmter Position in der Liste löschen
- Element anhand vom Wert aus Liste löschen

Damit lassen sich viele Aufgaben bei Listen schnell und einfach erledigen!

Cloud Command

Variablen und Strings
mit Listen arbeiten – Einsatz von Methoden

Über die Anweisung **dir(list)** erhalten wir alle Methoden zu Listen!

- Im Folgenden die Übersicht aller Methoden des Datentyps Liste:

Methode	Beschreibung
.append()	hinzufügen von einem Eintrag am Ende der Liste
.clear()	Löscht Einträge der Liste
.copy()	Erstellt eine Kopie
.count()	Liefert die Anzahl der vorhandenen Listeneinträge zurück
.extend()	Hinzufügen aller Einträge einer anderen Liste
.index()	Suche nach Element in Liste und liefert Index(Nummern fangen bei 0 an) zurück
.insert()	fügt einen Eintrag an der vorgegebenen Index-Nummer ein (folgende Einträge werden nach hinten verschoben)
.pop()	löscht den Eintrag aus der Liste des übergebenen Index und liefert dessen Inhalt als Rückgabewert
.remove()	löscht den Eintrag aus der Liste (im Gegensatz zu pop() wird der Wert und nicht der Index übergeben)
.sort()	sortiert die Liste in gewünschter Form