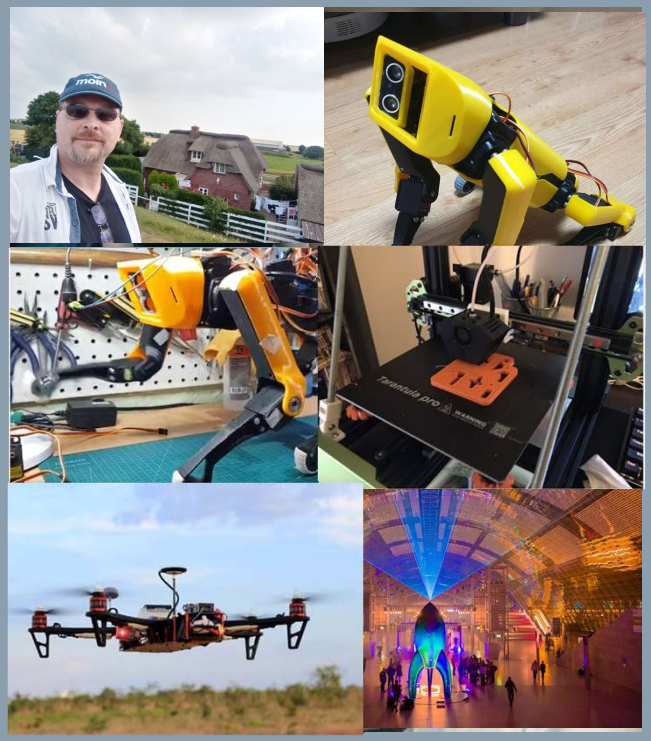
Cloud Command

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

Datentyp Dictionary – Wörterbücher in Python

- Der Datentyp Dictionary
  - (auf Deutsch „Wörterbuch“ bzw. assoziative Liste)
  - ermöglicht ein Aufbau ähnlich einer Liste aber als Indizes anstelle von Zahlen dann „Text“.
  - Der Begriff Dictionary ist wirklich mit dem dafür deutschen Wort „Wörterbuch“ am besten verständlich.
  - Über den Datentyp Dictionary können wir also in Python eine Zuordnungstabelle (in anderen Sprachen „assoziatives Array“) aufbauen und nutzen.
  - Durch die folgenden Beispiele wird es schnell verständlich.

Cloud Command

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

Schauen wir uns den Aufbau anhand einer deutsch-englisch Wörterbuches an. In anderen Programmiersprachen auch als „assoziatives Array“ bekannt.

- Wir wollen Zahlen in beiden Sprachen erhalten:
  - null = zero
  - eins = one
  - zwei = two
  - drei = three
- Als Dictionary wird in Python im ersten Schritt ein leeres „Wörterbuch“ erstellt
  - dies ist an den geschweiften Klammern zu sehen.
  - Wir nennen es „deutschenglisch“:

Cloud Command

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

Als Dictionary wird in Python im ersten Schritt ein leeres „Wörterbuch“ erstellt

- dies ist an den geschweiften Klammern zu sehen.
- Wir nennen es „deutschenglisch“
- Und füllen es mit unseren Daten

```
deutschenglisch = {}
deutschenglisch['null'] = 'zero'
deutschenglisch['eins'] = 'one'
deutschenglisch['zwei'] = 'two'
deutschenglisch['drei'] = 'three'
```

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

- Und wie üblich können wir den Inhalt ausgeben über

```
deutschenglisch
```

- Als Rückmeldung erhalten wir den kompletten Inhalt des Dictionary

```
>>> deutschenglisch
{'null': 'zero', 'eins': 'one', 'zwei': 'two', 'drei': 'three'}
```

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

Diese folgende Variante ist auch eine andere Möglichkeit, ein „Dictionary“ aufzubauen:

```
deutschitalienisch = {'eins': 'uno', 'zwei': 'due', 'drei': 'tre'}
```

- Wollen wir ein bestimmtes Element der assoziativen Liste erhalten, geben wir das gewünschte einfach mit dem „Klartext-Index“ an.

```
>>> deutschenglisch['zwei']
'two'
```

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

Löschen von Elementen - Aktion auf das dictionary

- Wollen wir ein Element, sprich Schlüssel-Wert Paar aus unserem Wörterbuch löschen, können wir dies direkt mit der Anweisung del und dem entsprechenden Index tun:

```
del deutschenglisch['zwei']
```

- Nun fehlt dieses Element in unserem dictionary

```
>>> deutschenglisch
{'null': 'zero', 'eins': 'one', 'drei': 'three'}
```

- Aber Vorsicht mit del!
- Wird kein Index (Schlüssel-Wert Paar) angegeben, wird das komplette dictionary gelöscht!

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

Aber Vorsicht mit del!

- Wird kein Index (Schlüssel-Wert Paar) angegeben, wird das komplette dictionary gelöscht!

```
del deutschenglisch
```

- Das kann also ins Auge gehen:

```
>>> del deutschenglisch
>>> deutschenglisch
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
NameError: name 'deutschenglisch' is not defined
```

Cloud Command

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

- Dictionary Methoden – Python stellt verschiedene Methoden bereit

```
>>> dir(dict)
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__',
'__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setitem__', '__sizeof__',
'__str__', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop',
'popitem', 'setdefault', 'update', 'values‘]
```

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

values()

- Über die Methode values() werden alle Werte des dictionary zurückgeliefert. Aus unserem vorherigen Wörterbuchbeispiel wäre das dann:

```
deutschitalienisch.values()
```

- Ergebnis:

```
>>> deutschitalienisch.values()
dict_values(['uno', 'due', 'tre'])
```

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

keys()

- Über die Methode keys() werden alle Schlüssel des dictionary zurückgeliefert. Aus unserem vorherigen Wörterbuchbeispiel wäre das dann:

```
deutschitalienisch.keys()
```

- Als Ergebnis dann:

```
>>> deutschitalienisch.keys()
dict_keys(['eins', 'zwei', 'drei'])
```

Cloud Command

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

items()

- Über die Methode items() werden alle Elemente des dictionary zurückgeliefert.
- Aus unserem vorherigen Wörterbuchbeispiel wäre das dann:

```
deutschitalienisch.items()
```

- Ergebnis:

```
>>> deutschitalienisch.items()
dict_items([('eins', 'uno'), ('zwei', 'due'), ('drei', 'tre')])
```

- Die Feinheiten der Rückgabe:
  - eckige Klammern zeigen, dass es sich um eine Liste handelt
  - runde Klammern zeigen, dass es sich um Tupel handelt

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

clear()

- Alle Elemente werden aus dem dictionary gelöscht. Im Unterschied zum vorher gezeigten del besteht weiterhin ein dictionary mit seinem Namen, dass allerdings keinen Inhalt hat.

```
deutschitalienisch.clear()
```

Cloud Command

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

copy()

- Legt eine Kopie des dictonary an.

```
woerterbuch = deutschitalienisch.copy()
```

- Nun steckt der Inhalt auch in dem neuen „dictionary“, dass aber diesen Inhalt so behält, auch wenn das Original verändert wird. Hier liegt der Unterschied zu einem Aliasnamen (siehe weiter unten).

```
>>> woerterbuch =
deutschitalienisch.copy()
>>> woerterbuch
{'eins': 'uno', 'zwei': 'due', 'drei': 'tre'}
```

- Wird mit Aliasnamen gearbeitet (und nicht mit Kopien) verändert sich der Inhalt, egal ob man mit dem Alias oder dem originalen Namen zugreift!

Variablen und Strings

Datentyp Dictionary – Wörterbücher in Python

- Wird mit Aliasnamen gearbeitet (und nicht mit Kopien)
  - verändert sich der Inhalt,
  - egal ob man mit dem Alias
  - oder dem originalen Namen zugreift!

```
neuernamen = deutschitalienisch
deutschitalienisch['vier'] = 'quattro'
neuernamen['vier']
```

- Wir erhalten als Ausgabe dann „quattro“.
- Die Änderungen gelten für alle Aktion, egal ob Elemente dazukommen oder entfernt werden.

Cloud Command

Variablen und Strings
Methoden von Dictionarys

- Über die Anweisung **dir(dict)** erhalten wir alle Methoden zu Dictionary

```
>>> dir(dict)
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__init__',
'__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce__',
'__reduce_ex__', '__repr__', '__setattr__', '__setitem__', '__sizeof__', '__str__',
'__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault',
'update', 'values']
```

- Im Folgenden die Übersicht aller Methoden des Datentyp Dictionary:

Variablen und Strings

Methoden von Dictionarys

Methode	Beschreibung
.clear()	Löscht alle Einträge des Dictionarys
.copy()	Erstellt eine Kopie
.fromkey()	Erstellt eine Kopie aus den Schlüsseln eines Dictionary
.get()	Liest einen Wert zu einem übergebenen Schlüssel aus
.items()	Gibt alles Schlüssel und Werte aus (bei Python 2.7 .viewitems())
.keys()	Zeit die Schlüssel eines Dictionary an
.pop()	löscht den Eintrag aus dem Dictionary des übergebenen Schlüssels und liefert dessen Inhalt als Rückgabewert
.popitem()	Liefert einen Eintrag als Tupel zurück und löscht den Eintrag aus dem Dictionary (im Gegensatz zu pop() muss kein Schlüssel übergeben werden)
.setdefault()	liefert den Wert eines Eintrags aus dem Dictionary, wenn der Schlüssel vorhanden ist. Ist kein entsprechender Schlüssel vorhanden, wird ein Schlüssel mit dem Wert im Dictionary gespeichert
.update()	Aktualisiert einen Wert, wenn der Schlüssel vorhanden ist. Wenn noch kein entsprechender Schlüssel vorhanden ist, wird Wert und Schlüssel eingetragen
.values()	Liefert alle Werte des Dictionary zurück (ohne Schlüssel)