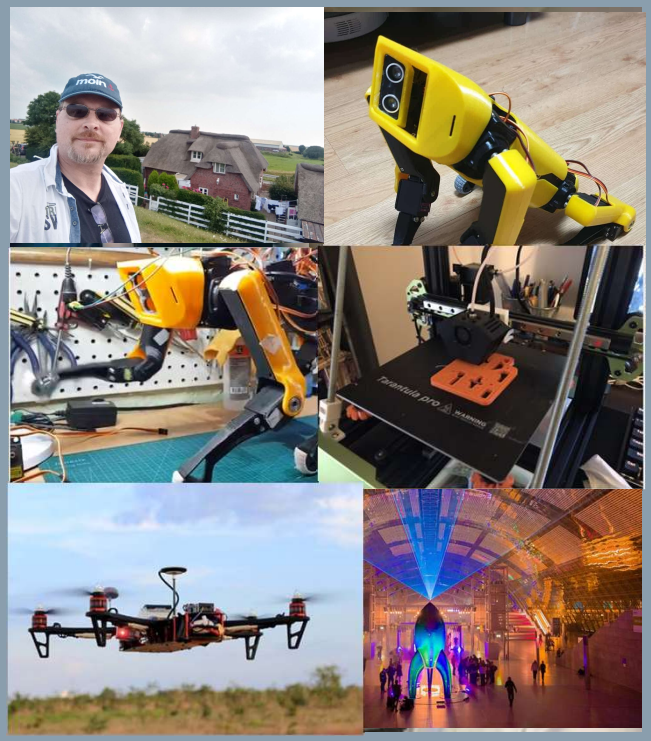
Cloud Command

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

Boolsche Operatoren
Grundlagen

- logische Operatoren (Boolesche Operatoren)

- Die logischen Operatoren werden gerne mit if-Abfragen genutzt.
- Im letzten Kapitel hatten wir Vergleich genutzt wie: „wert1 != wert2“
  - Bei den logischen Operatoren geht Python einen Schritt weiter
  - und kann Ausdrücke (Vergleiche) verketten.
  - Somit sind Abfragen möglich wie beispielsweise:
  - wert_1 == wert_2 ODER wert_2 > wert_3

Schreibweise Python	Bedeutung
wert_1 == wert_2	Ist gleich
wert_1 != wert_2	Ist ungleich
wert_1 < wert_2	Kleiner als
wert_1 <= wert_2	Kleiner oder gleich als
wert_1 > wert_2	Grösser als
wert_1 >= wert_2	Größer oder gleich als

Cloud Command

Boolsche Operatoren
Grundlagen

Hier haben wir die üblichen 3 booleschen Operatoren:

Befehl	Schreibweise Python	Bedeutung
and	wert_1 == wert_2 and wert_2 > wert_3	Und: sowohl wert_1 ist gleich wert_2 UND wert_2 ist größer wert_3
or	wert_1 == wert_2 or wert_2 > wert_3	Oder: entweder wert_1 ist gleich wie wert_2 ODER wert_2 ist größer als wert_3
not	not wert_1	Trifft zu, wenn wert_1 nicht gesetzt ist, also FALSE ist

Cloud Command

Boolsche Operatoren
Grundlagen

Bei den booleschen Werten hat man gerne Tabellen mit True/False für einen einfacheren Überblick:

x	y	X and y	X or y
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Cloud Command

Boolsche Operatoren
Grundlagen

- Wollen wir beispielsweise überprüfen, ob ein Wert gesetzt wurde, kann dies über den booleschen Operatoren not geschehen. Im folgenden Beispiel werden überprüft, ob wert_1 gesetzt ist:

```
if not wert_1:
    print("wert_1 ist False")
else:
    print("wert_1 ist True")
```

Cloud Command

Boolsche Operatoren
Grundlagen

- Natürlich könnte man auch hier ohne not arbeiten:

```
if wert_1:
    print("wert_1 ist True")
else:
    print("wert_1 ist False")
```

Cloud Command

Boolsche Operatoren
Grundlagen

- Aber in manchen Situationen ist nur wichtig zu überprüfen, ob False vorliegt:

```
if not wert_1:
    print("wert_1 ist False – bitte Eingabe machen!")
```

- Am Rande bemerkt:
  - Die Bezeichnung wurde zu Ehren von dem Mathematiker George Boole aufgrund seiner Erarbeitung der grundlegenden boolesche Algebra vergeben.
  - Lustigerweise wurde George Boole Mathematikprofessor ohne jemals selbst eine Universität besucht zu haben :).