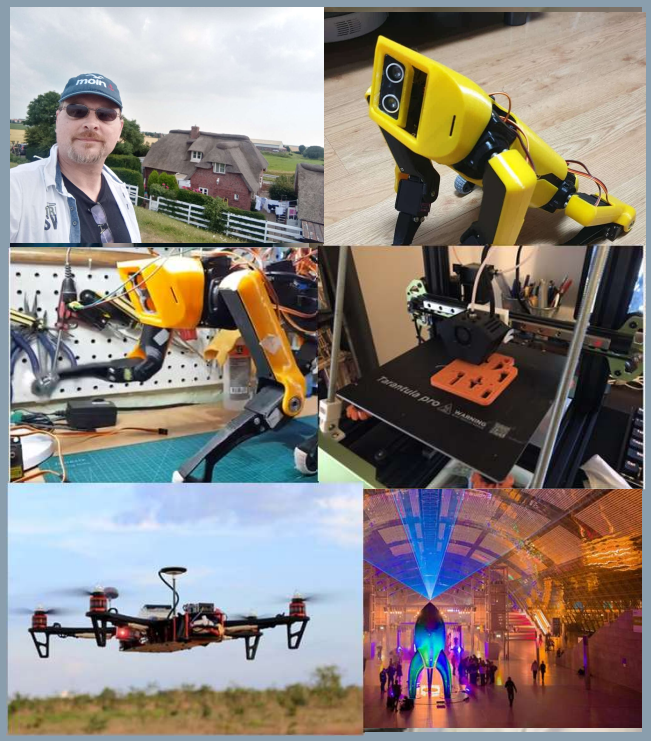
Cloud Command

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

Ausführbares Programm erstellen
Grundlagen

- Um ein ausführbares Python-Programm (Python-Script) zu erstellen, benötigen wir eine Datei.
  - Das Besondere an dieser Datei ist die Dateiendung.
  - Hier wird für Python dann ".py" verwendet.

- Es ist egal, mit welchem Betriebssystem wir arbeiten.
  - Im Beispiel wird unter Windows der im Betriebssystem vorhandene Editor „notepad“ verwendet.

- Die einfachste Variante ist, auf Start zu klicken und dann einfach loszutippen.
  - Sobald wir die ersten Buchstaben von „editor“ eingetippt haben bekommen wir bereits den Vorschlag für den Standardeditor.

- Diesen Editor auswählen.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

- Und nun können wir direkt im Editor unser Python-Programm eintippen.
- In unserem Texteditor geben wir ein:

```
print('Hallo Welt')
```

- Beim Speichern auf die Dateiendung „.py“ achten! Als Dateiname verwende ich für unser erstes Programm „hallowelt.py“.
- Ich speichere mein Python-Programm im Ordner „Lesson1“ unter „C:“ ab.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

Zum Starten des Python-Programms

- Python starten unter Windows
- Zum Starten des Python-Programms benötigen wir eine DOS-Konsole.
  - Diese rufen wir unter Windows auf, indem wir wieder auf den Start-Button klicken
  - und dort einfach „cmd“ eintippen.
  - Die vorgeschlagene DOS-Konsole können wir dann übernehmen.
- Ist bei Installation von Python nun der Pfad im Betriebssystem richtig hinterlegt worden, können in direkt in der DOS-Konsole Python starten.
- Zum Test kann man Python ohne unser geschriebenes Programm starten.
  - Einfach direkt in der DOS-Konsole „python“ eintippen.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

- Klappt das, können wir unser Python-Programm „hallowelt.py“ starten.
- Dazu werden nach dem Programmaufruf noch der Pfad und der Dateiname angegeben. Z.B.:

```
python c:\Lesson1\hallowelt.py
```

- Wir erhalten durch die Ausführung des Programms nun die Ausgabe in der Konsole von „Hallo Welt“.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

- Wenn wir uns den langen Pfadnamen zum eintippen sparen wollen,
  - können wir einmal in das Verzeichnis wechseln
  - über die DOS-Anweisungen cd , in das Verzeichnis „Lesson1“ wechseln
  - und dann das Programm ohne Pfadangaben ausführen.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

- Ausführen von Python-Programmen mit Linux oder dem Mac OSx

- Bei diesen starten wir das Terminal.
- Hier haben wir die Besonderheit, dass ggf.die alte Version von Python bereits im Betriebssystem ausgeliefert wird.
- Wir können direkt im Terminal python eingeben
  - und erhalten als Rückmeldung die Versionsnummer von Python
  - im Beispiel ist das „Python 2.7.10“.
  - Haben wir das aktuelle Python3 installiert,
    - können wir dieses in Terminal mit python3 aufrufen
    - und erhalten als Feedback die entsprechende Versionsnummer.

Cloud Command

Ausführbares Programm erstellen
Grundlagen

Cloud Command

Ausführbares Programm erstellen
Grundlagen

- Zum Starten von unserem Programm geben wir im Terminal nach dem Startbefehl für Python noch den Pfad und den Dateinamen an:

```
python3 Documents/Lesson1/hallowelt.py
# oder
python3 ~/Documents/Lesson1/hallowelt.py
```

- Das Programm wird dann ausgeführt.

Cloud Command

If - Bedingung
Grundlagen

if-Bedingung in Python

- In Python gibt es die Möglichkeiten Bedingungen zu überprüfen
  - und entsprechend im Programmablauf darauf zu reagieren.
- Hier könnten wir abhängig von der Uhrzeit den Nutzer entsprechend Begrüßen.
- Aber erst einmal der allgemeine Aufbau von if-Abfragen und wie wird diese einsetzen.

Cloud Command

If - Bedingung
Grundlagen

Aufbau der if-Abfrage

- Würden wir in Deutsch eine Bedingung formulieren, würde diese wie folgt aussehen
- wenn Wert kleiner als 5 dann mache alles Doppelpunkt
- Und nun das Ganze als Python-Programm, in der wir als Erstes die Variable definieren:

```
wert = 3
if wert < 5:
    print('Wert ist kleiner als 5')
```

- Das wichtige ist, dass nach der if-Abfrage das weitere, was zu der if-Abfrage gehört, eingerückt wird!
- Diese Einrückungen sind der Dreh- und Angelpunkt in Python und garantiert einen sauberen Quellcode.
- Pfuscht man bei den Einrückungen, funktioniert das Programm nicht wie erwartet.

Cloud Command

If - Bedingung
Grundlagen

- Bei unserer if-Anfrage gehört alles Folgende, was eingerückt ist, zu der Bedingung.
- Und das kann mehr als eine Zeile sein!

```
wert = 3
if wert < 5:
    print('Wert ist kleiner als 5')
    print('Ich gehöre auch noch zu der Bedingung‘)

print('und hier geht es nach der if-Abfrage weiter')
```

- Lassen wir das Programm ausführen, erhalten wir als Ergebnis alle 3 Zeilen ausgegeben.
  - Die 2 Textzeilen, die „innerhalb“ der if-Abfrage sind (sprich, die eingerückt sind)
  - und die letzte Textzeile, die nach der if-Abfrage kommt und nicht eingerückt ist.

Cloud Command

If - Bedingung
Grundlagen

- Ändern wir nun unsere Variable am Anfang auf **wert = 9**",
  - erhalten wir bei der Programmausführung nur noch die letzte Zeile ausgegeben, die nach der if-Abfrage kommt.
  - Die if-Abfrage ist nicht wahr,
    - weil der Wert mit 9 bereits größer ist als die Bedingung < 5
    - und somit das eingerückte der if-Abfrage nicht ausgeführt wird.

- Wichtig ist bei Programmänderungen, dass man speichern nicht vergisst!

Cloud Command

If - Bedingung
Grundlagen

- Was passiert eigentlich genau bei wert < 5? Python überprüft, ob das Ergebnis wahr oder falsch ist.
  - Dabei kann auch direkt „true“ oder „false“ der if-Abfrage präsentiert werden
  - und diese reagiert darauf entsprechend:

```
if True:
    print('if-Bedingung ist wahr')
```

- Es erscheint als Ergebnis:

```
if-Bedingung ist wahr
```

- Natürlich könnten wir auch eingeben if false:
  - und es würde nichts angezeigt werden,
  - da unsere if-Abfrage nicht wahr ist.

Cloud Command

If - Bedingung
Grundlagen

- Natürlich würde kein Mensch so eine if-Abfrage erstellen,
- denn die if-Abfrage würde ja immer exakt zum gleichen Ergebnis führen.
- Allerdings können wir auch Variablen übergeben, die diesen Wert haben.

```
wahroderfalsch = True
if wahroderfalsch:
    print('if-Bedingung ist wahr')
```

- Unsere Variable "wahroderfalsch" hat nun den Wert "true" zugewiesen bekommen
  - und die if-Abfrage reagiert entsprechend darauf.

- Diese Variablen sind vom Typ Boolean
  - es gibt nur die Werte true oder false bei Boolean.
  - Diese Variablen-Typen sind benannt nach dem Erfinder George Boole.

Cloud Command

If - Bedingung
Grundlagen

Ungleich != in Python

- Oft möchte man auch einfach wissen, ob eine bestimmte Bedingung nicht zutrifft, also ungleich ist.
- Dazu kann man den Operator **!=** (ungleich) nutzen.
- Als Beispiel wollen wir von einer Stundenzahl wissen, ob es NICHT 12 Uhr ist.
- Dazu unser Code:

```
wert = 11
if wert != 5:
    print('Es ist nicht 12 Uhr‘)
```

Cloud Command

If - Bedingung

Grundlagen

Alternative, wenn if-Abfrage nicht zutrifft

- Jetzt ist es oft geschickt gleich darauf reagieren zu können, wenn die if-Abfrage nicht zutrifft.
- Ohne diese Kenntnis müssten wir solche kantigen Konstruktionen bauen wie folgt:

```
wert = 9
if wert < 5:
    print('Wert ist kleiner als 5')

if wert > 4:
    print('Wert ist größer als 4')
```

Cloud Command

If - Bedingung

Grundlagen

- Dies wollen wir kürzer, da die zweite if-Abfrage einfach das Gegenteil gerade von der ersten darstellt.
- Alles was nicht kleiner ist als 5 ist auf jeden Fall größer als 4. Und hier kommt das schöne Wort else zum Einsatz:

```
wert = 9
if wert < 5:
    print('Wert ist kleiner als 5')
else:
    print('Wert ist größer als 4')
```

- Unseren zweiten Part ersetzen wird durch else:.
  - Trifft unsere if-Abfrage nicht zu, sprich ist diese nicht wahr, sondern falsch,
    - dann wird der Block unter else: ausgegeben

Cloud Command

If - Bedingung
Grundlagen

- Unseren zweiten Part ersetzen wird durch **else:**
- Trifft unsere if-Abfrage nicht zu, sprich ist diese nicht wahr, sondern falsch,
  - dann wird der Block unter else: ausgegeben
- Als Ausgabe erhalten wir nun bei der Programmausführung:

```
Wert ist größer als 4
```

Cloud Command

If - Bedingung

Grundlagen – IF Operatoren

Alle Vergleichs-Operatoren

- Je nach Aufgabenstellung den passenden Vergleich nutzen!

Operator	Beschreibung
==	gleich
!=	Ungleich
<	Kleiner
>	Größer
<=	Kleiner oder gleich
>=	Größer oder gleich

Cloud Command

If - Bedingung
Grundlagen

Weitere Bedingungen innerhalb der Bedingung prüfen – elif

- Unser bisheriges Programm ist nicht wirklich sexy.
- Es gibt die Ausgabe „Wert ist kleiner als 5“ oder „Wert ist größer als 4“.
- Eigentlich wären folgenden 3 mögliche Ergebnisse deutlich schicker.
  - Wert ist kleiner 5
  - Wert ist exakt 5
  - Wert ist größer 5

- Dazu brauchen wir eine Abfrage innerhalb der Abfrage.
  - Und dazu kennt Python den Befehl **elif**.
- In den meisten anderen Programmiersprachen kennt man dies als „elseif“
  - aber in Python ist es elif.

Cloud Command

If - Bedingung

Grundlagen

- Unser Programm von oben sieht nun damit so aus.

```
wert = 9
if wert < 5:
    print('Wert ist kleiner als 5')
elif wert == 5:
    print('Wert ist exakt 5')
else:
    print('Wert ist größer als 5')
```

- Der Überprüfung von der Bedingung hinter elif wird nur dann überprüft,
  - wenn die erste Bedingung nicht zutrifft.
  - Trifft unsere Bedingung unter elif auch nicht zu, dann kommt die Alternative unter else zum Tragen.

- Wichtig sind die Einrückungen!

Cloud Command

If - Bedingung
Grundlagen

Kleine böse Fehlerquelle – doppeltes Gleichzeichen

- Hier unser Code, in dem sich ein Fehler eingeschlichen hat:

```
wert = 9
if wert < 5:
    print('Wert ist kleiner als 5')
elif wert = 5:
    print('Wert ist exakt 5')
else:
    print('Wert ist größer als 5')
```

- Wir überprüfen bei elif, ob dieses exakt 5 entspricht.
- Hier ist der Standardfehler, dass versehentlich nur ein Gleichzeichen (anstelle von 2) gemacht wurde.
- Das ist aber nicht möglich und somit kommt eine Fehlermeldung von Python in Form von „SyntaxError: invalid syntax“

Cloud Command

If - Bedingung
Grundlagen

- beliebig viele elif

- Wir sind bei elif nicht auf eines begrenzt.
- Von diesen weiteren Abfragen können beliebig viele gemacht werden.
    - Je nach Aufgabenstellung ist es sinnvoll.