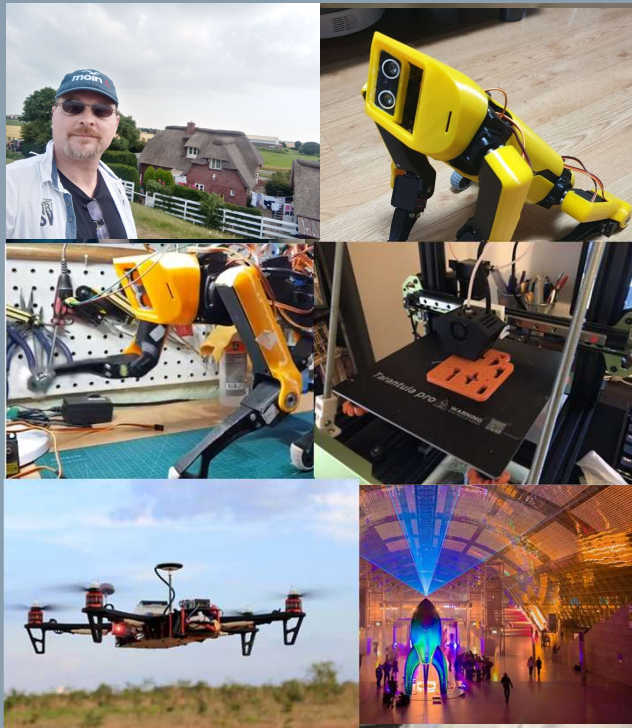


# Python Programmieren

## 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



## Variablen und Strings

### **Listen in Python – viele Inhalte geordnet speichern**

- Wir haben mit Variablen die Möglichkeit kennen gelernt, in einem Platzhalter (sprich die Variable) Inhalt zu speichern. Allerdings kann jede Variable nur einen Inhalt haben.
- Das kann je nach Fall unpraktisch werden.

## Variablen und Strings

### Listen in Python – viele Inhalte geordnet speichern

- Wenn man sich vorstellt, dass man ein Telefonbuch in Variablen speichern möchte, dann hätte man wilde Konstruktionen wie:

```
vorname1 = 'Axel'  
vorname2 = 'Elke'  
vorname3 = 'Martin'
```

- Aber wo soll das enden?
- Das wäre als eine wenig handliche Vorgehensweise.
- Und daher gibt es sogenannte „**Listen**“ in Python.
- Wer schon eine andere Programmiersprache kennt, hier wird diese Möglichkeit **Array** genannt.

## Variablen und Strings

# Listen in Python – viele Inhalte geordnet speichern

Wie können wir in Python nun in Listen Inhalte speichern?

- Ganz einfach über die eckigen Klammern

```
vornamen = ['Axel', 'Elke', 'Martin']
```

- Die Ausgabe kann wie gewohnt über print() erfolgen:

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

- Es werden nun alle Vornamen ausgegeben.

## Variablen und Strings

### Listen in Python – viele Inhalte geordnet speichern

- Wollen wir nun nur einen bestimmten Vornamen ausgeben, müssen wir die Index-Nummer mitgeben. Diese wird in einer eckigen Klammer geschrieben.

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

- Das Ergebnis der Ausgabe ist nun:

```
Elke
```

- Warum nicht das erste Element unserer Liste, was ja offensichtlich den Inhalt „Axel“ enthält?
- Hier ist es wichtig, dass Computer immer bei 0 anfangen zu zählen, insbesondere bei Listen.

## Variablen und Strings

### Listen in Python – viele Inhalte geordnet speichern

- Wollen wir also das erste Element unserer Vornamen-Liste erhalten, müssen wir als Index 0 angeben:

```
vornamen = ['Axel', 'Elke', 'Martin']  
print(vornamen[0])
```

- Jetzt bekommen wir das erste Element der Liste.
  - Daher der wichtige Merksatz
- 
- **MERKE:** Listenelemente fangen immer beim Index 0 an!

## Variablen und Strings

# Listen in Python – viele Inhalte geordnet speichern

### letztes Listenelement ansprechen

- Wir können auch von hinten anfangen etwas ausgeben.
  - Und hier fängt man mit -1 an!
  - Somit wird das letzte Element ausgegeben:

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

- Somit bekommen wir als Ausgabe unseren „Martin“.



## Variablen und Strings

# Listen in Python – viele Inhalte geordnet speichern

### Listenwerte überschreiben

- Wollen wir einen bestehenden Listeneintrag überschreiben, weil aus dem Martin eine Martina geworden ist, können wir das über die Index-Nummer und einer neuen Wertzuweisung machen:

```
vornamen = ['Axel', 'Elke', 'Martin']  
vornamen[2] = 'Martina'
```

- Lassen wir nun unsere Liste mit print() ausgeben:

```
vornamen = ['Axel', 'Elke', 'Martin']  
vornamen[2] = 'Martina'  
print(vornamen)
```

- kommt der Liste mit dem geänderten Wert:

```
['Axel', 'Elke', 'Martina']
```

## Variablen und Strings

# Listen in Python – viele Inhalte geordnet speichern

### Listen durch weitere Elemente erweitern

- Wir können Python-Listen einfach über das Pluszeichen erweitern.

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

- Nun wird die bestehende Liste um diese 2 Elemente erweitert. Lassen wir die Liste ausgeben, erhalten wir:

```
['Axel', 'Elke', 'Martina', 'Heike', 'Sabine']
```

- Die Schreibweise "vornamen += " ist die Kurzschreibweise von

- Python `vornamen = vornamen + ['neue Werte']` zur Verfügung.

## Variablen und Strings

### Listen in Python – viele Inhalte geordnet speichern

- Python stellt verschiedene Funktion für Listen zur Verfügung.
- Hier gibt es eine Funktion zum Erweitern von Listen:
  - `listenname.append('neuer Wert')`.
  - Als konkretes Beispiel sieht das nun so aus:

```
vornamen = ['Axel', 'Elke', 'Martin']  
vornamen.append('Rolf')
```

## Variablen und Strings

# Listen in Python – viele Inhalte geordnet speichern

### Komplette Liste löschen: del()

- Wird eine komplette Liste nicht mehr benötigt bzw. soll diese im nachfolgenden Code wieder neu aufgebaut werden, kann die alte Liste gelöscht werden. Dies geschieht über die Funktion del()
- Unser Beispiel von oben wandeln wir ab. Wir wollen nicht mehr die bestehende Liste mit den Vornamen erweitern, sondern durch einen komplett anderen Inhalt ersetzen:

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

- Als Ergebnis erhalten wir:

```
['Axel', 'Elke', 'Martina']  
['Heike', 'Sabine']
```