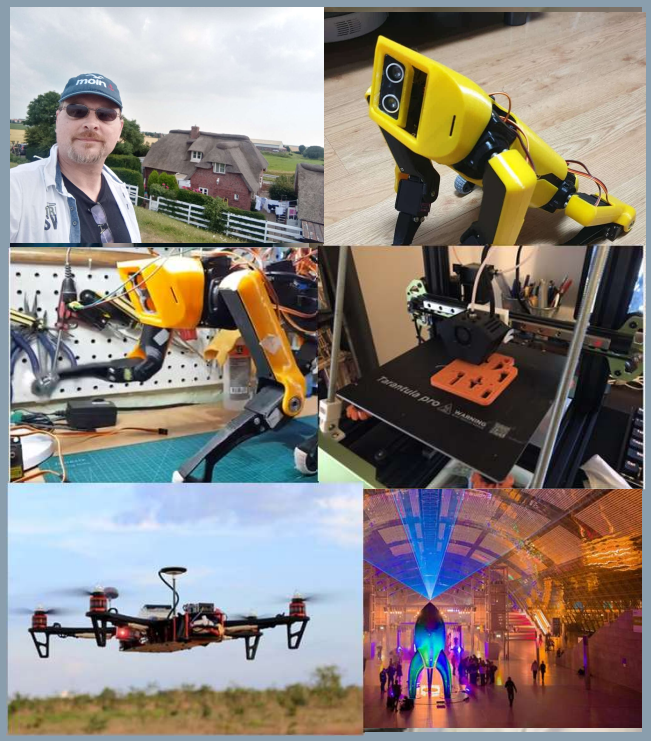
Cloud Command

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

Variablen und Strings

String Methode partition() – String in Einzelteile zerlegen

- Die Methode partition() erhalt als Parameter den Suchtext, anhand der komplette String zerlegt werden soll.

```
string.partition("Suchtext")
```

- Wir bekommen 3 Teile als Rückgabewerte:
  - alles vor dem Suchtext
  - den Suchtext
  - alles nach dem Suchtext

Variablen und Strings

String Methode partition() – String in Einzelteile zerlegen

- Schauen wir es am Beispiel an. Wir haben den Satz „Python ist einfach zu lernen“. Jetzt wollen wir diesen Text zerlegen, und zwar bei dem Wort „ist“.

```
satz = "Python ist einfach zu lernen"
ergebnis = satz.partition("ist")
print(ergebnis)
```

- Das Ergebnis ist als Datenform ein Tupel und wir erhalten:

```
('Python ', 'ist', ' einfach zu lernen')
```

- Bitte ein Augenmerk darauf, dass nichts verloren geht!
- Es werden auch alle Leerzeichen um der Suchwert „ist“ behalten.
- Dieser finden sich nach dem ersten Wert des Tupels nach „Python “ und vor dem dritten Wert „ einfach zu lernen“.

Variablen und Strings
String Methode partition() – String in Einzelteile zerlegen

Suchtext öfters vorhanden – was passiert?

- Was passiert nun eigentlich, wenn unser Suchtext öfters vorhanden ist? Erweitern wir unseren Satz auf: „Python ist einfach zu lernen und ist cool“.

```
satz = "Python ist einfach zu lernen und ist cool"
ergebnis = satz.partition("ist")
print(ergebnis)
```

- Im Ergebnis sieht man schön, dass nur das erste Auftreten des Suchtextes berücksichtigt wird. Das zweite „ist“ in unserem Beispiel endet im dritten Rückgabewert des Tupels.

```
('Python ', 'ist', ' einfach zu lernen und ist cool')
```

.

Cloud Command

Variablen und Strings
String Methode partition() – String in Einzelteile zerlegen

Suchtext am Anfang sofort vorhanden

- Was passiert, wenn der Suchtext sofort am Anfang unseres Strings vorhanden ist?

```
satz = "ist Python einfach zu lernen?"
ergebnis = satz.partition("ist")
print(ergebnis)
```

- Hier wird das Ergebnis für das Verständnis der Funktion von .partition() wichtig.
  - Als Rückgabe-Tupel erhalten wir:

```
('', 'ist', ' Python einfach zu lernen?')
```

- Unser Suchtext ist in dem Tupel also immer der zweite Wert! Tritt er im Text in der ersten Position auf, kommt vor ihm also nichts.
  - Somit bleibt der erste Wert unseres Rückgabetupels leer und im dritten Wert steht alles nach dem „ist“.

Variablen und Strings
String Methode partition() – String in Einzelteile zerlegen

Suchtext nicht vorhanden

- Wenn unser Suchtext nicht im Text vorkommt, ist die Reaktion der partition()-Methode, dass im Ergebnis alles im ersten Bereichs unseres Tupels landet und der zweite Wert des Tupels leer bleibt. Unsere Suche war ja nicht erfolgreich und im zweiten Wert steht immer das erfolgreich gesuchte Wort:

```
satz = "Ist Python einfach zu lernen?"
ergebnis = satz.partition("ist")
print(ergebnis)
```

- Ergebnis:

```
('Ist Python einfach zu lernen?', '', '')
```

- Man sieht an diesem Beispiel auch, dass Groß- und Kleinschreibung einen Unterschied macht.

Variablen und Strings

String Methode partition() – String in Einzelteile zerlegen

Fehlermeldung „ValueError: empty separator“

- Die Methode partition() benötigt einen Parameter (für den Suchtext), ansonsten erhält man als Fehlermeldung
  - „ValueError: empty separator“.
  - Das passiert, wenn ein leerer Suchtext übergeben wird:
    - ergebnis = satz.partition("").
- Gibt man anstelle des Suchtextes keinen Parameter ein
  - ergebnis = satz.partition(),
  - erhält man die Fehlermeldung: „TypeError: partition() takes exactly one argument (0 given)“

Variablen und Strings

String Methode partition() – String in Einzelteile zerlegen

Für das Aufspalten muss also ein sinnvoller Wert mitgegeben werden. Wobei ein Leerzeichen auch ein sinnvoller Wert ist:

```
satz = "Ist Python einfach zu lernen?"
ergebnis = satz.partition(" ")
print(ergebnis)
```

- Er wird dann einfach beim ersten Leerzeichen aufgeteilt. Unser Ergebnis:

```
('Ist', ' ', 'Python einfach zu lernen?')
```

- Die Methode partition() bietet viele Möglichkeiten im praktischen Einsatz. Einfach im Hinterkopf behalten, dass es diese Methode gibt.