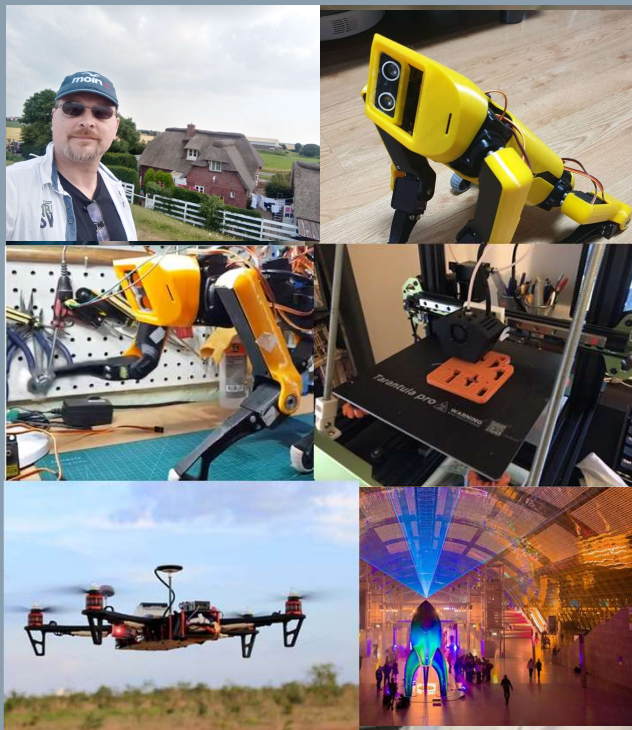


Python Programmieren

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

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

- Die Methode `.endswith()`
 - liefert „True“ zurück, wenn das gesuchte Ende vorhanden ist.
 - Ansonsten kommt als Rückgabewert „False“.
- Die englische Zusammenziehung liest sich ungewohnt.
- Übersetzt man es ein wenig „holprig“ ins deutsche, ist die Funktion der Methode klar: ends = wird beendet, endet / with = mit
- Also einfach die Methode „endet-mit“.

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

Syntax von `endswith()`

- `string.endswith(Suffix[, Startposition[, Endposition]])`
- Wir haben also 3 Parameter:
 - Suffix (muss angegeben werden): zu überprüfende Zeichenfolge
 - Startposition (optional): Startpunkt, ab dem überprüft wird
 - Endposition (optional): Endpunkt, bis zu dem überprüft wird

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

- Anhand eines Beispiels wird die Anwendung klarer.

```
inhalt = "https://cctrainer.ddnss.de"  
ergebnis = inhalt.endswith(".de")  
print(ergebnis)
```

- Wir bekommen als Ergebnis zurückgeliefert:

```
True
```

- Wir können also überprüfen, ob es zutrifft (hier in unserem Fall, ob es eine deutsche Domainendung ist).
- Damit können wir auch in eine if-Abfrage gehen oder in eine while-Schleife.

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

Anfangs- und Endposition einsetzen

- Je nach Fall ist es manchmal geschickt, die Anfangs- und Endposition für die Überprüfung festzulegen.
- Geben wir hier 28 als Ende an, bekommen wir weiterhin „True“ zurück, da unser String 28 Zeichen lang ist.

```
inhalt = "https://cctrainer.ddnss.de"  
ergebnis = inhalt.endswith(".de", 0, 26)  
print(ergebnis)
```

- Sobald wir 25 eingeben haben wir zur Überprüfung nur noch „...cctrainer.ddnss.d“ – und somit kein „.de“ mehr.

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

Mehrere Fälle überprüfen

- Diese Methode kann auch mit Tupel's eingesetzt werden. Hört sich kompliziert an, ist aber in der Praxis sehr einfach und wird öfters benötigt. Was ist ein Tupel? Ein Tupel ist eine Wertesammlung.
- Nehmen wir an, wir wollen unseren String überprüfen, ob die URL mit einer dieser Endungen endet:
 - `.de`
 - `.com`
 - `.net`

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

Also nehmen wir alle 3 Fälle (unsere Wertesammlung) auf und es soll, wenn einer der Fälle zutrifft „True“ zurückgeliefert werden.

- Wir erstellen aus unseren Fällen ein Tupel (siehe entsprechendes Kapitel <https://www.python-lernen.de/tupel.htm>)
- Unser Tupel aus den 3 Domainendungen:

```
datentyp_tupel = (".de", ".com", ".net")
```

- Dieses Tupel wird nun in unsere Methode `.endswith()` eingesetzt:

```
inhalt = "https://cctrainer.ddnss.de"  
datentyp_tupel = (".de", ".com", ".net")  
ergebnis = inhalt.endswith(datentyp_tupel)  
print(ergebnis)
```

- Jetzt kann die URL auf „.de“ oder auf „.com“ oder auf „.net“ enden und wir erhalten ein „True“ zurück.

Variablen und Strings

„Endet mit“-Methode: Strings auf Suffix überprüfen mit `.endswith()`

- Jetzt kann die URL auf „.de“ oder auf „.com“ oder auf „.net“ enden und wir erhalten ein „True“ zurück.
- Öfters wird man diese Konstruktion sehen. Hier ist schneller ersichtlich, dass ein Tupel eingesetzt wird:

```
inhalt = "https://cctrainer.ddnss.de"  
ergebnis = inhalt.endswith((".de", ".com", ".net"))  
print(ergebnis)
```