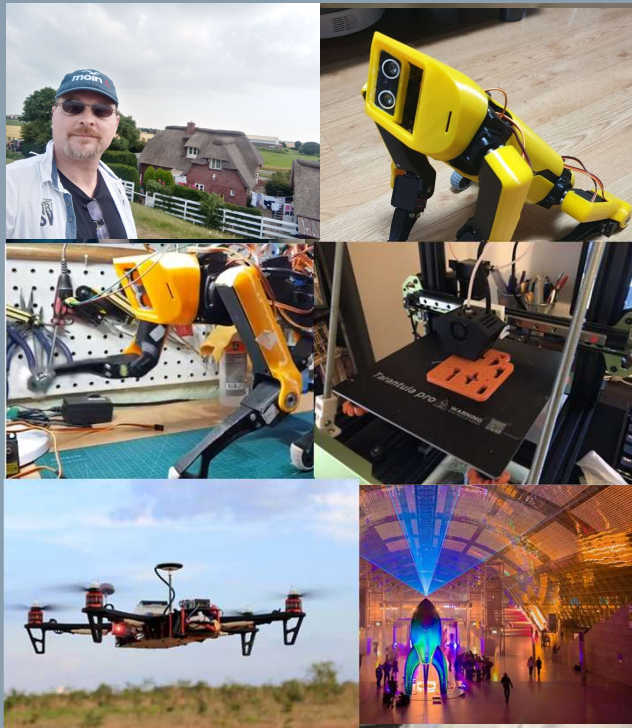


Python Programmieren

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



Zufallszahlen über random() Grundlagen

mit random Zufallszahlen nutzen – import random in Python

- Die meisten Spiele nutzen den Zufall für das Spiel.
- Als erstes einfaches Spiel programmieren wir „Schere, Stein, Papier“ in Python um die Anwendung von random in Python kennen zulernen.
- Um random nutzen zu können, müssen wir das random-Modul in unser Python-Programm importieren!

Zufallszahlen über random() Grundlagen

- Was ist ein Modul und warum gibt es dieses Konzept?
- Eigentlich könnte man meinen, dass eine Zufallsfunktion so wichtig ist, dass diese sofort verfügbar ist.
- Dies ist nicht der Fall:
 - diese Funktionen sind im „Lieferumfang“ der normalen Installation enthalten, müssen aber erst „aktiviert“ werden.
 - Dieses „Aktivieren“ erfolgt über das Importieren des Moduls.
 - So kann jedes benötigte Modul vom Programmierer nachgeladen werden und Python schlank und somit auch schnell gehalten werden.

Zufallszahlen über random() Grundlagen

Das „Aktivieren“ erfolgt über die Anweisung import:

```
import random
```

- Über die Hilfeanweisung erhalten wir die ersten Informationen zu verfügbaren Methoden der Anweisung random.

```
print(help(random.random))  
help(random.random)
```

- Als Rückmeldung erhalten wir:

```
random() -> x in the interval [0,1)
```

Zufallszahlen über random() Grundlagen

- Zufallszahlen zwischen 0 und 0.99999
- Wir bekommen also eine zufällige Zahl beim Aufruf von random() zurückgeliefert.
- Wer genau auf die Rückmeldung der Python-Hilfe schaut, dem fällt die eckige Klammer vor 0 und die runde Klammer nach 1 auf.
- Die eckige Klammer bedeutet,
 - dass eine 0 zurückgeliefert werden kann
- die runde Klammer bedeutet dagegen,
 - dass niemals eine 1 zurückgeliefert wird!

Zufallszahlen über random() Grundlagen

Lassen wir uns ein Ergebnis ausgeben:

```
print(random.random())
```

- Bei jeder Ausgabe erfolgt eine andere Zahl – das ist das Wesen des Zufalls.

```
>>> print(random.random())  
0.1229030667914014
```

```
>>> print(random.random())  
0.4683265429999067
```

```
>>> print(random.random())  
0.5755756960623487
```

Zufallszahlen über random() Grundlagen

durch uns festgelegtes Intervall für random

- Wenn wir nun Zufallszahlen in einem anderen Intervall benötigen, könnten wir eine eigene Funktion dafür programmieren. Das ist eine Möglichkeit. Hier der Vollständigkeit halber.
- Dadurch bekommen wir Zahlen zwischen 1 und unter 10

```
def zufallszahl():  
    return random.random() * 9 + 1  
print (zufallszahl())
```


Zufallszahlen über random() Grundlagen

random.uniform – eigenes Intervall einfach festlegen

- Python bietet über die Anweisung random.uniform weitere Möglichkeiten:

```
help(random.uniform)
```

- Wir bekommen über random.uniform eine zufällige Nummer zwischen „a“ und „b“ zurückgeliefert.

```
print(random.uniform(5,8))
```

- Somit bekommen wir nur noch Nummer zwischen 5 und maximal 8.

Zufallszahlen über random() Grundlagen

Zufallsnummer aus Normalverteilung (Glockenkurve)

- Zufallsnummer aus Normalverteilung
 - `random.normalvariate(Standardabweichung sigma)`

```
print(random.normalvariate(5, 0.1))
```

Zufallszahlen über random() Grundlagen

Würfel simulieren randint(min, max)

- Wenn wir wie bei einem Würfel eine festgelegte Anzahl von Möglichkeiten haben, die alle gleich wahrscheinlich sind, dann bietet sich randint an.

```
print(random.randint(1,6))
```

- Wir bekommen nur ganzzahlige Werte wie bei einem normalen 6-seitigen Würfel zurück zwischen eins und sechs (inklusive der Sechs).
- Gut sichtbar, wenn wir 15-mal den Würfel „werfen“:

```
for i in range(15):  
    print(random.randint(1,6))
```

Zufallszahlen über random() Grundlagen

Zufälliges Element aus einer Liste

- Wir können auch aus einer Liste von Elementen ein Element zufällig auswählen lassen. Hier sind wir unserem „Schere, Stein, Papier“.
- Wir erstellen also als Erstes eine Liste mit unseren 3 Elementen:

```
handgeste = ['Schere', 'Stein', 'Papier']
```

- Und nun können wir Python zufällig auswählen lassen:

```
print(random.choice(handgeste))
```

- Das Modul random bietet noch weitere Möglichkeiten, die einem über `print(dir(random))` angezeigt werden. Für die meisten Anwendungen reichen die bisher vorgestellten aus.
- Somit haben wir schon eine schöne Möglichkeit für die zufällige Auswahl der Handgeste. Im folgenden Kapitel bauen wir unser Spiel zusammen.