

# Hangman

## Python + SQL

# 1 Projekt

In diesem Projekt wollen wir das Spiel [Hangman](#) (Galgenmännchen) programmieren. Hierbei soll der Computer ein zufälliges Wort aus einer SQL-Datenbank (`database.db`) mit einer bestimmten Anzahl an Buchstaben zufällig auslesen. Der Spieler (in diesem Fall wir) soll daraufhin durch geschicktes Wählen einzelner Buchstaben, das gesuchte Wort erraten. Dafür hat er nur eine bestimmte Anzahl an Versuchen zur Verfügung (z.b. `MAX_VERSUCHE = 5`).

Zusätzlich wollen wir noch ein zweites Programm schreiben, mit dem wir zusätzliche Wörter zur Datenbank hinzufügen können.

## 1.1 Das Spiel

### 1.1.1 Die Datenbank

Die Datenbank `database.db` (die sich im Kursbereich befindet) enthält eine Tabelle (TABLE) mit dem Namen `words`. Diese Tabelle enthält 2 Spalten, `word` und `letters`:

- `word`: Ein Wort.
- `letters`: Die Anzahl der Buchstaben des Wortes

Hier ein möglicher Ausschnitt der Tabelle in tabellarischer Form:

word	letters
KATZE	5
BOOT	4
MAUS	4
HAUSBOOT	8
UHR	3
EURO	4
...	...
...	...

#### Wichtig!

Um unser Programm einfacher zu gestalten unterscheiden wir nicht zwischen Groß- und Kleinschreibung. D.h., dass jedes Wort in der Datenbank nur mit **Großbuchstaben** geschrieben wird, auch neue Wörter, die in die Datenbank geschrieben werden!

### 1.1.2 Teilaufgaben

Wir stellen fest, dass wir unterschiedliche Funktionalität für unser Spiel benötigen.

- Mit `sqlite3` eine Verbindung zur Datenbank aufbauen
- Eine Anfrage an die Datenbank stellen, Wörter einer bestimmten Länge zurückzugeben.
- Von diesen Wörtern *eines* zufällig auswählen
- Den Spieler die Möglichkeit geben, das Wort über die Konsole zu erraten.
- Das Spiel beenden, wenn der Spieler mehr Versuche benötigt als ihm erlaubt sind.

Um eine Verbindung zur Datenbank aufzubauen, kann folgender Code helfen:

```
import sqlite3
connection = sqlite3.connect("pfadzurdatenbank.db")

cur = connection.cursor()
```

Über das Objekt `cur` können nun Anfragen an die Datenbank gestellt werden ( z.b. "gibt mir alle Wörter mit X Buchstaben zurück"). Schaue im Zweifelsfall zur Erinnerung auch nochmals die Lektionen zu `sqlite` im Kursbereich an.

Möglicherweise benötigst du an einem bestimmten Punkt die Funktionalität aus einer Liste ein zufälliges Element auszuwählen. Hierbei hilft dir möglicherweise folgender Code :

```
# Ein Element aus einer Liste per Zufall selektieren
from random import choice

a = choice([1,2,3])
print(a)
```

### 1.1.3 Beispiel

Hier ein mögliches Resultat für das Spiel. Der verwendete Text kann sich natürlich in deinem Programm unterscheiden. Wichtig ist, dass die Funktionalität gewährleistet ist.

```
Eine neue Runde Hangman!
Wähle die Anzahl der Buchstaben, welche das Wort enthalten soll.
Oder gib EXIT ein, wenn Spiel beendet werden soll.
3
Ein Wort wurde ausgewählt...
Die Raterunde beginnt!
Du hast 5 Versuche, das Wort rauszufinden!
Viel Spaß!

____
Welchen Buchstaben wählst du?
A
A ist leider nicht vorhanden....

____
Verbleibende Anzahl Versuche: 4
Welchen Buchstaben wählst du?
E
E ist leider nicht vorhanden....

____
Verbleibende Anzahl Versuche: 3
Welchen Buchstaben wählst du?
U
U_
Verbleibende Anzahl Versuche: 3
Welchen Buchstaben wählst du?
R
U_R
Verbleibende Anzahl Versuche: 3
Welchen Buchstaben wählst du?
H
UHR
Geschafft!
Eine neue Runde Hangman!
Wähle die Anzahl der Buchstaben, welche das Wort enthalten soll.
Oder gib EXIT ein, wenn Spiel beendet werden soll.
2
Keine Wörter gefunden...
Eine neue Runde Hangman!
Wähle die Anzahl der Buchstaben, welche das Wort enthalten soll.
Oder gib EXIT ein, wenn Spiel beendet werden soll.
3
Ein Wort wurde ausgewählt...
Die Raterunde beginnt!
Du hast 5 Versuche, das Wort rauszufinden!
Viel Spaß!

____
Welchen Buchstaben wählst du?
E
E ist leider nicht vorhanden....

____
Verbleibende Anzahl Versuche: 4
Welchen Buchstaben wählst du?
I
I ist leider nicht vorhanden....

____
Verbleibende Anzahl Versuche: 3
Welchen Buchstaben wählst du?

```

## 1.2 Kommunikation mit der Datenbank

### 1.2.1 Beschreibung

Wir wollen ein zusätzliches kleines Programm schreiben, welches uns ermöglicht über die Konsole neue Wörter in die obige Datenbank zu schreiben. Dadurch können wir den Pool der Wörter, welche ausgewählt werden, vergrößern.

**Wichtig!**

Beachte, dass alle Wörter automatisch in **Großbuchstaben** in die Datenbank geschrieben werden sollen!

**Wichtig!**

Beachte außerdem, dass die Anzahl der Buchstaben des Wortes *auch* in die Tabelle der Datenbank geschrieben werden müssen!

### 1.2.2 Beispiel

Ein mögliches Resultat könnte wie folgt aussehen:

```
Welches Wort soll der Datenbank hinzugefügt werden (oder EXIT)?
Lastwagen
Das Wort LASTWAGEN wurde der Datenbank hinzugefügt
Welches Wort soll der Datenbank hinzugefügt werden (oder EXIT)?
python
Das Wort PYTHON wurde der Datenbank hinzugefügt
Welches Wort soll der Datenbank hinzugefügt werden (oder EXIT)?
Fenster
Das Wort FENSTER wurde der Datenbank hinzugefügt
Welches Wort soll der Datenbank hinzugefügt werden (oder EXIT)?
Baum
Das Wort BAUM befindet sich schon in Datenbank!
Welches Wort soll der Datenbank hinzugefügt werden (oder EXIT)?
EXIT
```

Wenn wir das Spiel jetzt erneut spielen, ist es möglich, dass wir die neuen Wörter erraten müssen!

**Abgabe**

Die fertigen Programme sollen als reiner Pythoncode geschrieben werden (also mehrere Dateien mit der Endung \*.py).