

## Twist:

## Lösungsidee:

Am besten werden zur Lösung der Aufgabe zwei Funktionen geschrieben, eine, die ein Wort twistet und eine zum Enttwisten eines Wortes. Beim Twisten wird der erste und letzte Buchstabe unberührt gelassen und die inneren werden durcheinandergewürfelt. Beim Enttwisten wird nach einem Wort gesucht, das den gleichen Anfangs- und Endbuchstaben sowie die gleiche Länge des zu enttwistenden Wort hat. Außerdem muss die Häufigkeit jedes Buchstabens bei beiden Wörtern gleich sein. Hierfür wird die bereitgestellte Liste häufiger deutscher Wörter verwendet.

## Umsetzung:

Die Lösungsidee wird in Python implementiert. Es wird die Funktion `shuffle()` aus dem Modul `random` benötigt. Zuerst wird eine Liste mit häufig verwendeten Satzzeichen erstellt und die bereitgestellte Liste mit den deutschen Wörtern in ein dictionary übertragen, nach Anfangsbuchstaben sortiert. Danach wird die vom Nutzer eingegebene Datei geladen und in die Variable `text` eingetragen, sie ist eine Liste, in der jeder Eintrag für eine Zeile steht, diese sind wiederum Listen mit allen Wörtern der entsprechenden Zeile.

Darauf folgt eine Funktion, die ein Wort übernimmt und es getwistet wieder ausgibt: Hat das Wort weniger als vier Buchstaben, wird es so, wie es ist übernommen. Sonst werden dem Wort die Satzzeichen vor und hinter ihm entfernt, sie werden später wieder hinzugefügt. Der erste und letzte Buchstabe bleibt ebenfalls, so wie er ist, deshalb wird er entfernt und später mit den Satzzeichen wieder hinzugefügt. Der übrige Teil des Wortes wird nun „geschufflet“ und bekommt seine Satzzeichen, Anfangs- und Endbuchstaben zurück.

Die nächste Funktion nimmt ein getwistetes Wort entgegen und gibt es enttwistet wieder zurück: Gesucht ist ein Wort aus der dictionary, das das getwistete Wort sein könnte. Zuerst wird die Groß- und Kleinschreibung ignoriert (sie wird später wieder hinzugefügt). Es werden alle Wörter aus dem dictionary überprüft, die den gleichen Anfangs- und Endbuchstaben sowie die gleiche Länge haben. Nun wird getestet, ob jeder Buchstabe in dem zu enttwistenden Wort in dem aktuellen Wort aus der dictionary in der richtigen Häufigkeit vorkommt. Wurde kein Wort gefunden, das passt, wird das getwistete Wort wieder zurückgegeben.

Darauf folgt ein Codeteil, der den Text Zeile für Zeile und Wort für Wort mithilfe der oben beschriebenen Funktion twistet und den getwisteten Text über die Konsole ausgibt. Daraufhin wird der gesamte Text enttwistet: Da Wörter mit weniger als vier Buchstaben nicht getwistet wurden, werden sie eins zu eins übernommen. Außerdem werden Satzzeichen entfernt und die Groß- und Kleinschreibung, die in der Funktion entfernt wird, vermerkt. Im letzten Teil des Codes wird jedes Wort enttwistet (mithilfe der Enttwist-Funktion), großgeschrieben, wenn dies vorher der Fall war, und mit den Satzzeichen verbunden. Jede enttwistete Zeile wird nun als letzter Schritt ausgegeben.

## Beispiel:

Wird z.B. verlangt, dass der Code den Text `twist2.txt` twistet und wieder enttwistet, wird erst der getwistete Text ausgegeben:

„Hat der alte Hxnsteemier sich dcoh emainl webgegeben! Und nun sellon siene Gesiter auch nach meniwm Weilln leebn. Sinee Wort und Werke mkret ich und den Bracuh, und mit Geätskristee tu ich Wnuder auch.“

Und danach der möglichst gut enttwistete:

„Hat der alte Hexenmeister sich doch einmal webgegeben! Und nun sollen seine Geister auch nach meinem Willen leben. Seine Wort und Werke merkt ich und den Brauch, und mit Geistesstärke tu ich Wunder auch.“