

1. **Titel:** Maßgeschneiderte Rätsel mit Python erstellen – `wortlabyrinth.py` als Beispiel maschinell generierter Knobelaufgaben.
2. **Vorhaben-Projektbeschreibung** Dieses Projekt dient als Beispiellösung, einen gewünschten Rätseltyp automatisiert zu erstellen. Die Generierung des Rätsels erfolgt mit Hilfe eines selbst geschriebenen Python-Skripts (`wortlabyrinth.py`). Das fertige [Skript](#) ist auf GitHub frei verfügbar. Ziel soll es sein, eine Sammlung gleichartiger Skripte kollektiv zusammenzutragen, damit für Lehrer*innen der hohe Arbeitsaufwand für die Erstellung and Anpassung solcher Knobel-Aufgaben drastisch reduziert wird, und gleichzeitig eine höhere Individualisierung bei der Aufgabenerstellung möglich wird.
3. **Ausgangslage und Bedarf:** Dieses Projekt erwuchs aus dem Wunsch, passgerechtere eigene Arbeitsmaterialien möglichst zeiteffektiv für den Schulunterricht erstellen zu können, gepaart mit einer persönlichen Affinität zu Aufgabentypen mit einer spielerischen Komponente.
Knobel- und Denkaufgaben haben ein großes Potential problemorientiertes Denken unser Schülerinnen und Schüler zu fördern. Darüber hinaus kann eine intrinsische Motivation für beliebige Lerninhalte und Klassenstufen erreicht werden, sofern die Rätsel von den Kindern als anspruchsvoll, aber nicht als zu schwierig empfunden werden. Die Krux: die Erstellung and Anpassung solcher Knobel-Aufgaben ist je nach Komplexität mitunter sehr zeitaufwendig. Es gibt mittlerweile für einige Aufgabentypen (z.B. Kreuzwörter und Suchsel) kostenfreie Generatoren, allerdings nur mit eingeschränkter Kontrolle über das fertige Endprodukt (den Output).
Kopiervorlagen mit fertigen Rätseln existieren zwar, passen aber selten genau zum gewünschten Lerninhalt oder haben einen ungeeigneten Schwierigkeitsgrad. Die Nutzung eigener Skripte ermöglicht eine bessere Individualisierung sowie leichtere Differenzierung. Durch Einbeziehung von Zufallsvariablen kann man beliebig viele Varianten der gleichen Aufgabe mit minimalen Aufwand generieren.

4. **Ziele des Projekts:** Ich möchte mit diesem Projekt Kolleginnen und Kollegen, die über Programmierkenntnisse verfügen (oder diese gerne erlangen würden), dazu animieren eigene Skripte für die Erstellung von Unterrichtsmaterial zu entwickeln und diese mit anderen zu teilen.

Des weiteren möchte ich betonen, dass die Programmierung des Skripts auch mir persönlich einen großen Lernzuwachs erbracht hat. Ein Computer-Programm zu schreiben bedeutet eigene Lösungsansätze für Probleme zu suchen und (bestenfalls) zu finden. Es war eine kurzweilige Denksportaufgabe, das Durchhalten lohnt sich!

5. **Verlauf des Projekts:** Zu Beginn war die Frage zu klären, in welcher Programmiersprache das Skript entstehen soll. Ich habe mich für Python entschieden, weil Python eine vielseitig einsetzbare höhere Programmiersprache ist, die plattformunabhängig und weit verbreitet ist. Eine aktive Gemeinschaft unabhängiger und gemeinnütziger Software-Entwickler verbessern und erweitern Python ständig. Es gibt mittlerweile ein beachtliches Repertoire an offenen (opensource) Skripten, die beispielsweise auf der Software-Entwicklerplattform GitHub (>70 Millionen Nutzer) zu finden sind.

Python erfreut sich besonders in technischen und naturwissenschaftlichen Arbeitsfeldern einer hohen Beliebtheit. Gerade für das Generieren kleiner Programme, die der Automatisierung bestimmter Vorgänge dienen, ist Python gut geeignet. Zudem ist Python eine relativ Beginner-freundliche Computersprache und es gibt eine Reihe hilfreicher Lern-Ressourcen (z.B. [Codeacademy](https://www.codecademy.com/)), um sich Python anzueignen.

Grundlegende Programmierkenntnisse (Datentypen, Definitionen, Funktionen, etc.) sind für die Umsetzung eines solchen Projekts absolut erforderlich. Diese können sich aber zügig angeeignet werden. Man lernt viel durch Ausprobieren (Trial and Error).

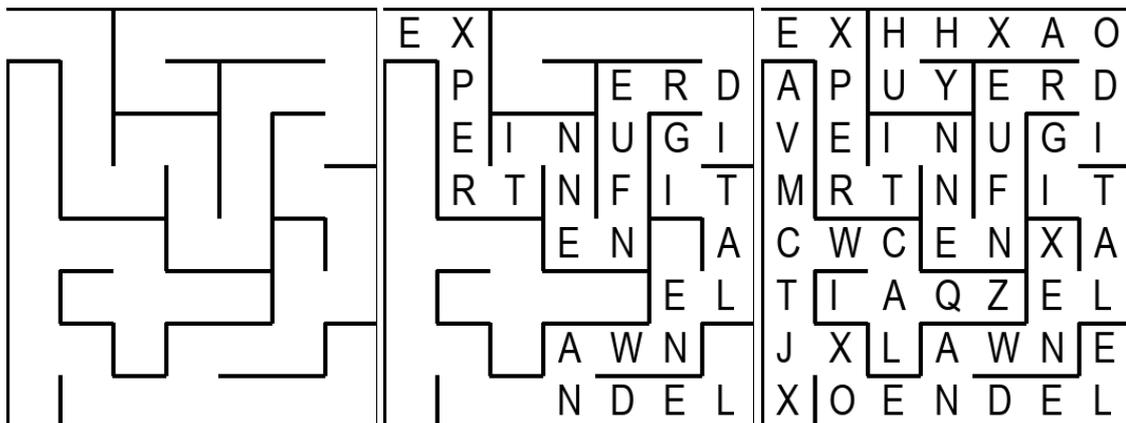
Programmierung ist ein iterativer Prozess. Man beginnt damit den Vorgang (was das Programm erledigen soll) in kleinere Aufgabenpakete aufzuteilen. In diesem Fall:

- Abfrage eines Lösungssatzes, der als Pfad im Labyrinth abgebildet werden soll
- Generieren von Labyrinth definierter Größe mit eindeutiger Lösung
- Auswahl eines zufällig generierten Labyrinths mit geeigneter Lösungspfad-Länge (damit der Lösungssatz eingesetzt werden kann)
- „Schreiben“ des Lösungssatzes in den Lösungspfad
- Füllen aller leeren Felder mit zufälligen Buchstaben

Durch die Segmentierung ist das Erreichen von Teilzielen möglich, was sich motivierend auf den Schaffensprozess auswirkt. Wenn Hilfe nötig ist sollte man sich nicht davor scheuen andere Menschen mit Erfahrung zu befragen (sei es anonym in einem Forum oder privat bei Bekannten).

Nach der Fertigstellung eines funktionalen Skripts folgt eine Phase des ausgiebigen Testens, welches des Ausmerzen von Programmfehlern (Debugging), sowie die Code-Optimierung (z.B. schnellere Programm-Laufzeiten) beinhaltet.

6. **Wesentliche Ergebnisse und Ausblick:** Durch den Prozess konnte ich ein besseres Verständnis für die Entwicklung von Computer-Programmen gewinnen (von der Konzeption bis zur Umsetzung). Das Resultat ist ein Labyrinth-Generator, den man auf verschiedene Lerninhalte anwenden kann.



Das Skript befindet sich momentan noch in der Testphase. Folgende Verbesserungen am Python Programm sind geplant:

- Umschreiben des Codes nach PEP-8 (Programmier-„Knigge“ für Python)
- verbesserte Eingabemaske (momentan über Kommando-Zeile)
- Überdenken/Änderung des Algorithmus zur Erstellung des Labyrinths, sodass mehr Sackgassen entstehen (besser für Differenzierung nach Leistung)
- ästhetische Optimierungen (Konstantsschrift, durchgängige Linien der Labyrinth-Wände, Vektorgrafik)

Ich habe vor weitere Skripte der gleichen Art online auf GitHub zu sammeln und dort verfügbar zu machen. Sofern sich weitere Kolleg*innen finden, die ähnliche Interessen verfolgen, kann man über eine geeignete Möglichkeit nachdenken, die Skripte zu katalogisieren und an einem Ort zu bündeln. Gerne in Zusammenarbeit mit dem LISUM.

7. **Unterstützungssysteme:** Ich konnte von der Expertise eines langjährigen Schulfreundes profitieren, der viel Erfahrung im Bereich Informatik mitbringt. Von ihm stammen viele Anregungen zur Verbesserung des Skripts. Darüber hinaus gibt es eine Reihe hilfreicher Onlineresourcen (Bsp.: <https://www.python.org/>, <https://www.learnpython.org/>), wobei viele in Englisch sind.
8. **Erkenntnisgewinn:** Meine drei wichtigsten Erkenntnisse:
- (1) Der initiale Aufwand lohnt sich – wenn das Programm einmal läuft, hat man ein sehr anpassungsfähiges Stück Software (und sammelt das nötige Know-How um weitere Skripte zu schreiben)
 - (2) Funktion über Form – wichtig ist, dass das Programm läuft wie gewünscht. Über alles Weitere kann man sich im Anschluss Gedanken machen.

- (3) Software-Entwicklung ist ein Prozess. Am Ergebnis findet man nahezu in jedem Fall Verbesserungspotential. Dies sollte jedoch nicht entmutigen, sondern im Gegenteil: dazu beflügeln, weitere Verbesserungen vorzunehmen.
9. **Tipps aus der Praxis für die Praxis:** Meine fünf Empfehlung für alle, die Ähnliches umsetzen wollen:
- (1) Du brauchst zum Programmieren kein Geld investieren – nutze kostenfreie Ressourcen! (egal ob Internetseite mit Onlinekurs, Datei-Editor oder integrierte Entwicklungsumgebung)
 - (2) Sei experimentierfreudig – viele Probleme lösen sich von selbst, indem man Dinge ausprobiert (learning by doing)
 - (3) Gehe kleinschrittig vor! Sonst verlierst du beim Programmieren den Überblick und die Fehlersuche wird unnötig schwer.
 - (4) Kommentiere deinen Code aus! Man kann sich später sonst nur noch schwer an Details erinnern.
 - (5) Hol dir weitere Meinungen und Ideen von anderen Personen ein! Nichts ist schlimmer als im eigenen Saft zu schmoren.
10. **Feedback:** Ich konnte das Programm bislang nur im eigenen Unterricht erproben. Erste Rückmeldungen fielen positiv aus. Kolleg*innen zeigten sich überrascht und neugierig.
11. **Materialien / Links / Anhänge:**
Python Skript auf Github: <https://github.com/jtypmel/wortlabyrinth> (inklusive GNU 3.0 Lizen)
12. **Meine Kontaktdaten.**
Dr. Jan Typmel
Goethe-Oberschule Kremmen
Straße der Einheit 2
16766 Kremmen (Brandenburg)
j.typmel@goethe-oberschule-kremmen.de

