

# Programmierung eines Vokabeltrainers

Untertitel

# Inhaltsverzeichnis

---

Einleitung .....	4
Anforderungen und Zielgruppe .....	4
Aufbau .....	6
Client-Server-Modell .....	6
Datenspeicherung .....	6
Wie lerne ich am besten .....	9
Wiederholung macht den Meister.....	9
Die magische Zahl $7 \pm 2$ .....	9
Pomodoro-Technik.....	10
Was man schreibt, das bleibt .....	11
Funktionen .....	12
Qt-Framework .....	13
Was ist ein Framework.....	13
Funktion .....	13
PyQt .....	13
Beispiel eines Miniprogrammes.....	14
Signale und Slots .....	15
Objektorientierte Programmierung.....	15
QtDesigner .....	16
Ausblick.....	17
x.....	17
Anhang .....	18
Schlusswort .....	18
Quellen .....	20
Hilfsmittel und Bibliotheken .....	21



## Einleitung

---

Man kann dem Umstand nicht aus dem Weg gehen, Vokabeln lernen gehört zum Alltag eines jeden Schülers. Niemand macht es gerne, aber man muss durch. Da stellt sich die Frage, wie man mit möglichst wenig Zeitaufwand seinen Wortschatz effizient und nachhaltig erweitern kann.

Früher haben wir vermutlich alle noch mit analogen Karten aus Papier und einem Karteikasten gelernt. Mit der Zeit haben die meisten Quizlet für sich entdeckt. Quizlet macht vieles leichter. Der Vorteil dieser Webseite ist es, dass man sich bereits von anderen Benutzern erstellte Lernsets von Wörtern herunterladen kann und dabei fast jedes Lehrmittel auf der Seite abgedeckt ist. Auch sonst muss nur eine Person das Set erstellen und kann es danach mit allen anderen teilen.

Quizlet war übersichtlich, hatte alle nötigen Funktionen ohne Schnick-Schnack, es war alles, was man brauchte. Leider liegt hier die Betonung auf «war». Wie es so häufig ist, sobald ein Produkt beliebt ist, fängt das dahinterstehende Unternehmen an, den Gewinn zu maximieren. So waren mit der Zeit immer mehr Funktionen nur noch mit Quizlet Plus, der bezahlten Version von Quizlet, verfügbar. Aber vor allem wurde Quizlet unübersichtlich und voller Ablenkungen. Es gibt unnötige Lernspiele, nervige Werbung und KI-Assistenzen, welche vom eigentlichen Kern ablenken: Möglichst zeitsparend Vokabeln lernen.

Das hat mich bereits vor 2 Jahren aufgeregt. Ich habe damals gerade gelernt mit Office-Makros, das sind kleine Programme in einem Office Dokument, zu programmieren. Deshalb habe ich direkt in der Excel-Datei, in der wir die Englisch-Wörter bekamen, begonnen ein Vokabeltrainer zu programmieren. Das Projekt wuchs und der «Vocitrainer» unterstützt inzwischen fünf Sprachen, bietet drei Lernmodi an und hilft mir mit der Aussprache des Wortes direkt im Programm.

Allerdings hat dieser Vokabeltrainer aus verschiedenen Gründen, wie einer mangelhaften Wartung von Seite Microsofts, keine grosse Zukunft. Ich habe immer wieder mit dem Gedanken gespielt, den «Vocitrainer» in der inzwischen erlernten Programmiersprache 'Python' von Grund auf neu zu schreiben. Als ich in der sechsten Klasse begonnen habe, mir Gedanken über meine Maturaarbeit zu machen, kam schnell die Idee auf, diese Gedanken in die Tat umzusetzen.

## Anforderungen und Zielgruppe

Da meine Ressourcen für das Projekt begrenzt sind, ist es wichtig, klare Anforderungen zu setzen und andere «unnötige» Funktionen wegzulassen.

- 1) Mir ist es wichtig, dass es mit dem neuen Vokabeltrainer möglich ist, ganz unkompliziert selbst erstellte Lernsets mit anderen zu teilen, indem man sie auf einen öffentlichen

'Marketplace' hochlädt. Jeder andere Benutzer kann diesen durchsuchen und sich Lernsets herunterladen.

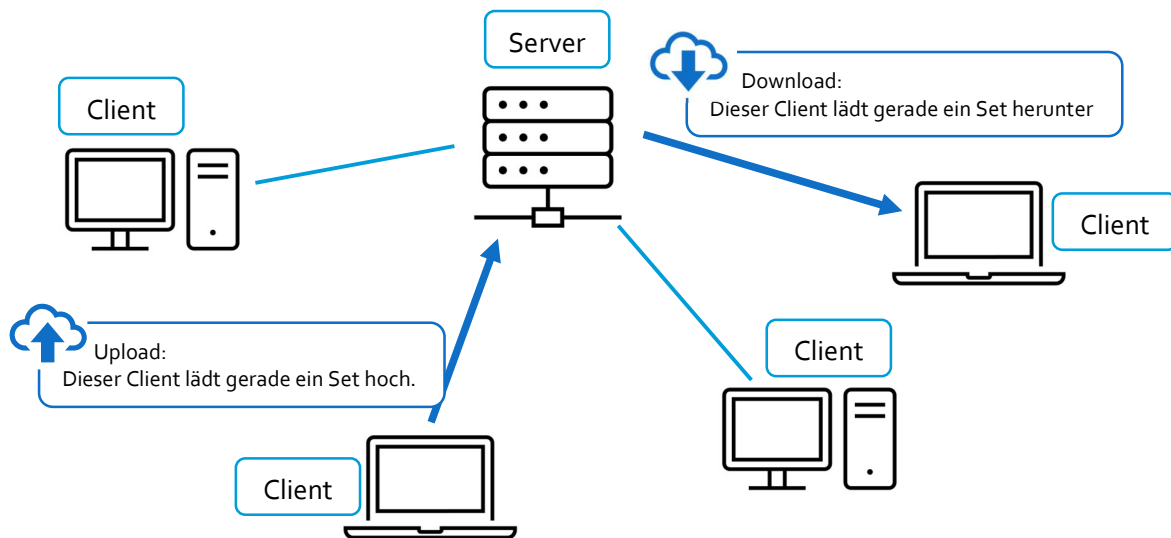
- 2) Da aber nicht von Anfang an viele Lernsets bereits zu Verfügung stehen werden, ist es mir wichtig, dass man ganz einfach die Lernsets, welche man zum Beispiel in Form von CSV-Dateien schon hat, importieren kann. Optimal wäre ein Tool, dass Lernsets direkt von Quizlet herunterlädt.
- 3) Die wichtigste Eigenschaft eines Vokabeltrainers ist es, dass man damit effizient lernen kann. Gerade weil man in der Praxis nicht regelmässig lernt, sondern erst die paar Tage vor der Prüfung, ist es wichtig, dass der Algorithmus ein schnelles Lernen fördert. Deshalb versuche ich, Erkenntnisse aus der Lernpsychologie und der Funktionsweise des Gehirns direkt in den Abfragealgorithmus einzubauen.
- 4) Meine Erfahrung ist es, dass Webapplikationen zur Ablenkung verführen, da sie im Browser offen sind und damit nur ein bis zwei Klicks von den neusten Nachrichten und kleinen Onlinespielen entfernt sind. Da Ablenkung für das Gehirn den Lerneffekt massiv reduzieren, gilt es das Potenzial dafür möglichst zu minimieren. Deshalb und weil ich bereits Python kann, habe ich mich für ein typisches Computerprogramm entschieden, welches offline nutzbar ist.

Die Zielgruppe sind Benutzer, die sich nicht mit den üblichen Lösungen zufriedengeben, Ablenkung minimieren und Effizienz maximieren wollen: Der typische computeraffine «Poweruser».

# Aufbau

## Client-Server-Modell

Das Client-Server-Modell ist ein Konzept aus der Informatik, um Aufgaben aufzuteilen. Der Server ist ein Computer, der wie ein Dienstleister fungiert. Er hat Informationen, welche von jedem Kunden (Client) angefordert werden können. Deshalb ist es auch ein entscheidendes Merkmal des Servers, dass dieser immer läuft.



Dieses Konzept hilft mir dabei, einen Marketplace, wie in den Anforderungen beschrieben, umsetzen zu können. Die Clients sind dabei die Schüler, mit ihrem Programm. Wenn nun beispielsweise eine Schülerin ein neues Set von Vokabeln mit ihren Mitschülern teilen will, lädt ihr Programm dieses im Hintergrund auf den Server. Der Server-Computer speichert das auf seiner Festplatte ab. Wenn jetzt die anderen Mitschüler das Set herunterladen wollen, schickt ihr Programm im Hintergrund eine Anfrage «Bitte schicke mir das Set 'xy'» an den Server. Dieser antwortet mit dem Set, welches dann im Vokabel Trainer angezeigt wird.

## Datenspeicherung

Bei einem Vokabel Trainer gibt es viele Daten, welche gespeichert werden müssen. Dabei müssen sowohl Daten auf dem Client, wie auch Daten auf dem Server gespeichert werden. Beim Benutzer auf dem PC werden die ganzen Vokabeln und Sets gespeichert, sowie die Ordner Struktur (welches Lernset in welchem Ordner ist). Der Server muss sich zusätzlich noch die verschiedenen Benutzer und ihre Passwörter merken.

Um all diese Informationen zu speichern gibt es verschiedene Möglichkeiten. Ich könnte die Daten mit einer eigenen Struktur in einer einfachen Textdatei speichern. Alternativen dazu wären Formate wie JSON<sup>1</sup>-Dateien und XML<sup>2</sup>-Dateien, welche bereits eine definierte Struktur haben.

Ich habe mich für eine Datenbank entschieden. In einer Datenbank kann man auch Daten strukturiert speichern, aber sie bietet noch weitere Vorteile.

1. Die fixe Struktur von Datenbanken erlaubt es nicht, dass in der Datenbank Unordnung herrscht (Konsistenz und Integrität)
2. Eine Datenbank kann ich einfach sortieren oder durchsuchen. Dazu gibt es eine eigene Abfragesprache: SQL<sup>3</sup>.
3. Datenbanken sind für grosse Datenmengen und viele Benutzer optimiert und verbrauchen deshalb weniger Ressourcen.
4. In Datenbanken können Beziehungen zwischen den Daten gespeichert werden. Damit wird Redundanz vermieden. Der Fachbegriff dazu ist «relationale» Datenbank.

Stark vereinfacht kann man sie Datenbanken wie eine grosses Excel-Dokument vorstellen. Wie auch in einem solchen Dokument, gibt es bei einer Datenbank unterschiedlicher Tabellen, in welchen verschiedenartige Informationen gespeichert werden. In der Datenbank des Clients gibt es verschiedene Tabellen. Sie speichern die Heruntergeladenen Vokabeln (Tabelle «karte») und wie diese in Sets und Ordnern organisiert sind (Tabellen «vociset» und «ordner»)



Karte_ID	Wort	Fremdwort	...	Lernfortschritt	Set_ID
1	Apfel	apple	...	3	1
2	Pfirsich	peach	...	2	1
3	blau	bleu	...	1	2

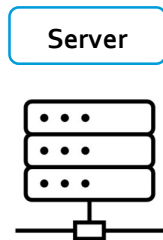
Set_ID	Name	Sprache	...	Ordner_ID	Ordner_ID	Name	...	Urordner_ID
1	Fruits	Englisch	...	1	1	Unit 3	...	0
2	Couleurs	Französisch	...	2	2	Unité 1	...	0

<sup>1</sup> JavaScript Object Notation: Ein menschenlesbares Datenformat

<sup>2</sup> Extensible Markup Language: Ein weiteres verbreitetes Datenformat

<sup>3</sup> Structured Query Language

Auf dem Server werden auch Karten und Vokabel-Sets gespeichert, deshalb gibt es auch dort die Tabellen «karte» und «vociset». Da es auf dem Server keine hierarchische Ordnerstruktur gibt, in der die Vokabel Sets gespeichert werden, braucht es keine Tabelle «ordner.» Dafür gibt es eine Tabelle «user», in welcher die verschiedenen Benutzer mit E-Mail, Passwort und Benutzername gespeichert werden.



Karte_ID	Wort	Fremdwort	...	Set_ID
1	Apfel	apple	...	1
2	Pfirsich	peach	...	1
3	blau	bleu	...	2

User_ID	Benutzername	E-Mail	...	Passwort-Hash
1	KSZler	meie.karl.2020@ksz...	...	f89840gaoafe...
2	Schüler16	martin436@gmail.com	...	14f766749559...

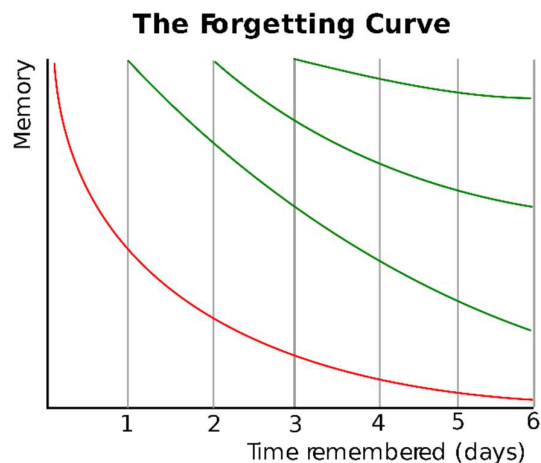
Set_ID	Name	Sprache	...	Downloads	User_ID
1	Fruits	Englisch	...	56	
2	Couleurs	Französisch	...	32	



# Wie lerne ich am besten

## Wiederholung macht den Meister

Ende des 19. Jahrhundert untersuchte der deutsche Psychologe Herman Ebbinghaus den Einfluss von Wiederholungen auf den Lerneffekt. Er fand heraus, dass das Lernen effektiver ist, wenn die Lerninhalte über einen längeren Zeitraum verteilt werden. Seine bekannteste Arbeit ist die Vergessenskurve, welche wohl die meisten Menschen schon mal gesehen haben. Die rote Linie zeigt die Abnahme des verbleibenden Wissens über die Zeit. Wenn man jedoch nach einem Tag die Lerninhalte nochmals wiederholt, vergisst man nicht mehr so schnell (1. Grüne Kurve) Mit jeder Wiederholung wird die Kurve flacher. Nach mehrmaligem Wiederholen sinkt allerdings der Verbesserungseffekt (Die Kurve flacht kaum mehr ab). Überlernen lohnt sich also nicht, da es nur kleine Vorteile bringt.



## Die magische Zahl 7 ±2

In diesem Kapitel geht es um die von George Miller beschriebene 'magische Zahl'. Ich habe einen Selbstversuch vorbereitet, mit dem man versuchen kann, die eigene 'magische Zahl' zu finden. Auf den nächsten Seiten befinden orangene Kästen, in denen der Selbstversuch beschrieben ist. Achtung: Vorblättern verfälscht das Ergebnis!

### Selbstversuch

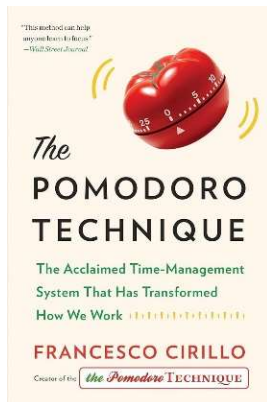
**Schritt 1:** Auf der nächsten Seite findest du einen weiteren orangen Kasten mit 20 Wörtern. Präge dir die Wörter gut ein, ohne sie aufzuschreiben! Du hast 1 Minute Zeit.

Der 1956 von George A. Miller veröffentlichte Artikel «The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information» gehört zu den meistzitierten Artikeln der Psychologie. Darin beschreibt Miller den Umstand, dass wir uns nur ungefähr 7 Informationseinheiten im Kurzzeitgedächtnis merken können.

Diese Erkenntnis hat konkrete Folgen darauf, wie wir am effizientesten neue Wörter lernen. Ein Wort zu erlernen ist ein Prozess. Wenn man es zum ersten Mal sieht, versucht man sich die

Bedeutung einzuprägen, beim zweiten bzw. dritten Mal versucht man bereits die richtige Schreibweise zu lernen. Während diesem Prozess sollte das Wort im Kurzzeitgedächtnis bleiben. Miller's Law angewendet heisst also, dass nur immer 7 Vokabeln auf einmal in diesem Lernprozess sein dürfen. Sind es mehr, nimmt der Lernerfolg rapide ab: Das Wort wird aus dem Kurzzeitgedächtnis verdrängt, bevor es wieder abgefragt und damit gefestigt wird.

## Pomodoro-Technik



Häufig fällt es uns schwer, uns zu konzentrieren. Ablenkungen sind überall und viele leiden am Drang, auch nur kurz das eigene Social Media zu checken. Dem Italiener Francesco Cirillo ging es ähnlich. Deshalb hat er die Pomodoro-Technik entwickelt. Er nahm eine Küchenuhr (in Tomatenform, ital. Pomodoro) und stellte sie auf 10 Minuten, während diesen wollte er

voll fokussiert arbeiten. Später hat er die Technik weiterentwickelt und ein Buch darüber geschrieben (siehe Bild).

Die Technik funktioniert folgendermassen:

1. Man stellt sich eine Aufgabe
2. Eine Uhr auf 25 Minuten einstellen
3. 25 Minuten fokussiert durcharbeiten
4. Alles stehen lassen und 5 Minuten Pause machen
5. Das kann man mehrmals wiederholen. Nach vier 'Pomodori' sollte man eine grössere Pause von bis zu 30 Minuten einlegen.

Die Idee ist, dass man mit einem klar definierten Ziel und beschränkter Zeit besser konzentrieren kann, ohne sich ablenken zu lassen.

Natürlich wird das zuerst nicht gelingen, aber das schöne ist, die Pomodoro-Technik lässt sich trainieren. Und dank den klaren Einheiten, lässt sich leicht messen, wie produktiv man war.

### Selbstversuch

*Haus*

*Quadrat*

*Tür*

*Buch*

*Apfel*

*Auto*

*Schuh*

*Kugel*

*Band*

*Schlüssel*

*Fisch*

*Box*

*Kopf*

*Hammer*

*Pfeil*

*Katze*

*Blume*

*Milch*

*Baum*

*König*

## Selbstversuch

Schritt 2: Schreiben Sie nun ohne nachzuschauen alle Wörter, an die sie sich erinnern können, auf ein Blatt Papier. Sie haben dafür etwa eine halbe Minute Zeit.

## Was man schreibt, das bleibt

Eine neuere Studie von 2014 namens «The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking» zeigt, dass das handschriftliche Schreiben Vorteile gegenüber dem Tippen hat. Auch wenn die Studie vor allem untersuchte, wie die Unterschiede beim Notizen machen von Videos sind, leite ich daraus einiges für das Vokabellernen ab.

1. Da das handschriftliche Schreiben länger dauert, hat das Gehirn mehr Zeit, das Geschriebene zu verinnerlichen.
2. Der Akt des Schreibens mit der eigenen Hand hat ein vertieftes Lernen zur Folge.
3. Und speziell für das Abschreiben von Vokabeln: Gerade bei schwierigen Wörtern hilft es, dass wir mit dem Blick mehrmals zwischen dem Wort und dem selbst geschriebenen wechseln, was den Lerneffekt weiter verbessert.

Die Frage ist nun, wie sich diese Tatsache in einen digitalen Vokabel-Trainer einbinden lässt. Natürlich kann man es nicht erzwingen, aber wenn ein Wort zum ersten Mal gezeigt wird, zeigt der Vokabeltrainer eine Aufforderung, das Wort einmal handschriftlich abzuschreiben.

### **Selbstversuch**

Schritt 3: Zähle, wie viele Wörter sie richtig haben. Vermutlich konnten sie sich an 5-9 Wörter erinnern. Diese Limitierung ist es, welche «Millers Law» beschreibt.

## Funktionen

---

## Qt-Framework

---

Das Qt-Framework (Qt ausgesprochen wie das Englische 'cute') ist wohl die wichtigste Komponente meiner Arbeit. Ich habe viel Zeit gebraucht, um dieses umfangreiche Framework auch nur im Ansatz kennen zu lernen. Deshalb möchte ich ihm ein eigenes Kapitel widmen und erklären, wie Qt im Grundsatz funktioniert. Es gibt auch ein kleines Tutorial, wie man als Anfänger eine kleines Programm schreiben kann.



Dieses Kapitel ist sehr technisch und besonders für die Leser, die Python bereits kennen und interessiert sind, wie all diese grafischen Benutzeroberflächen funktionieren.

### Was ist ein Framework

Framework nennt man eine grosse Sammlung von bereits geschriebenen Programmen. Diese kümmern zum Beispiel darum, wie ein Fenster genau auf dem Bildschirm gezeichnet wird und wie man das dem Betriebssystem sagt. Frameworks sind essentiell, weil sie einem viel Arbeit abnehmen. Es ist nicht möglich, für jedes kleine Projekt alles von Grund auf zu programmieren.

Im Unterschied zu einer Bibliothek, die eine ähnliche Funktion hat, bestimmt ein Framework auch die Struktur des Programmes und setzt eine definierte Anwendung voraus.

### Funktion

Qt ist ein Framework für grafische Benutzeroberflächen auch GUI (Graphical User Interface) genannt. Das Framework ist sehr verbreitet, da es einen grossen Funktionsumfang hat, Open Source ist (jeder kann es verwenden) und in vielen Programmiersprachen verfügbar ist (siehe Absatz [PyQt](#)).

Qt wird unter anderem vom legendären VLC Media Player verwendet wie auch von: Adobe Photoshop Elements, Google Earth, VirtualBox, Spotify (Linux Version) oder dem Simulationsprogramm der ESA.<sup>1</sup> Auch alles was man bei meinem Vokabeltrainer sieht ist mit Qt umgesetzt

### PyQt

Qt ist, wie viele Bibliotheken für Python in C++ geschrieben, da diese Programmiersprache einfach viel schneller und «genauer» ist und sich deshalb besser dafür eignet. Das heisst aber auch, dass ich Qt in Python nicht einfach so verwenden kann. Es braucht ein Glied zwischen Python und dem in C++ programmierten Qt, das die beiden verbindet. Das nennt man einen Wrapper. Für Qt gibt es 2 Python-Wrapper. PyQt und PySide. Es gibt keine grossen Unterschiede zwischen den beiden; ich habe mich für PyQt entschieden. PyQt implementiert über 1000 Klassen von Qt.<sup>2</sup>

## Beispiel eines Miniprogrammes

Das ist ein Beispiel eines minimalistischen PyQt-Fensters.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QPushButton

# Erstellen einer Qt-Anwendung [1]
app = QApplication(sys.argv)

# Erstellen eines Fensters [2]
window = QWidget()
window.setWindowTitle('Ein PyQt-Fenster')
window.setGeometry(100, 100, 300, 100) # Position und Größe des Fensters

# Erstellen eines Labels mit Text "Hallo Welt" [3]
label = QLabel('Hallo Welt', window)

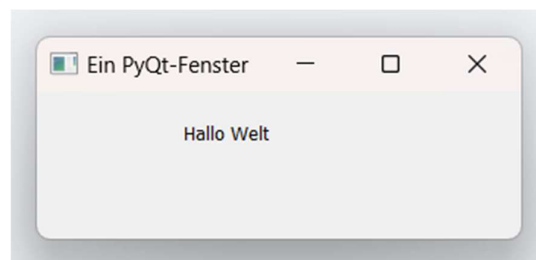
# Position des Labels im Fenster festlegen
label.move(100, 20)

# Fenster anzeigen
window.show()

# Anwendung ausführen [4]
sys.exit(app.exec_())
```

Zuerst erstellen wir eine Qt-Anwendung [1]. Dann wird das eigentliche Fenster initialisiert [2]. Dabei kann jedes Widget ein Fenster sein. Es gibt aber auch extra Klassen dafür wie QDialog oder QMainWindow. Nun fügen wir dem Fenster ein einfaches Label hinzu [3]. Als letztes starten wir das Programm [4]. Ab dann läuft die Event Loop, bis das Programm geschlossen wird.

Wenn wir das Programm ausführen, erscheint diese Fenster:



## Signale und Slots

Eine wichtige Frage von Gui-Anwendungen ist immer, wie werden Benutzereingaben verarbeitet? Qt hat hier sein eigenes Konzept namens Signals & Slots.

Elemente können in Qt bei einer Aktion ein Signal senden. Zum Beispiel ein Button der geklickt wird. Diese Signale können mit einem Slot verbunden werden. Dieser Slot ist meistens eine selbst geschriebene Funktion, in welcher man dann die Eingabe verarbeitet.

In unserem Beispiel von oben könnten wir Folgendes ergänzen.

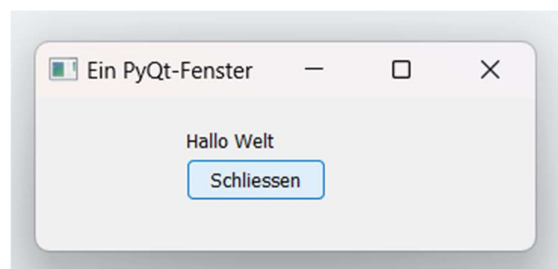
```
# Slot für das Schliessen Signal des Buttons [1]
1 usage
def fenster_schliessen():
    window.close()

# Einen Schliessen Button einfügen
cmd_schliessen = QPushButton("Schliessen", window)
cmd_schliessen.move(100, 40)
cmd_schliessen.clicked.connect(fenster_schliessen) # [2]
```

Wir erstellen eine Funktion, die das Fenster schliesst [1] und verbinden diese als Slot mit dem 'clicked'-Signal des Buttons, den wir erstellt haben [2].

Sobald man nun auf diesen neuen Button klickt, wird das Fenster geschlossen.

Wer das Beispiel das wir bis jetzt betrachtet haben weiter in Gedanken zu einem grösseren Projekt ausweitet, merkt, dass das schnell unübersichtlich werden kann. Deshalb brauchen wir zwei Hilfsmittel.



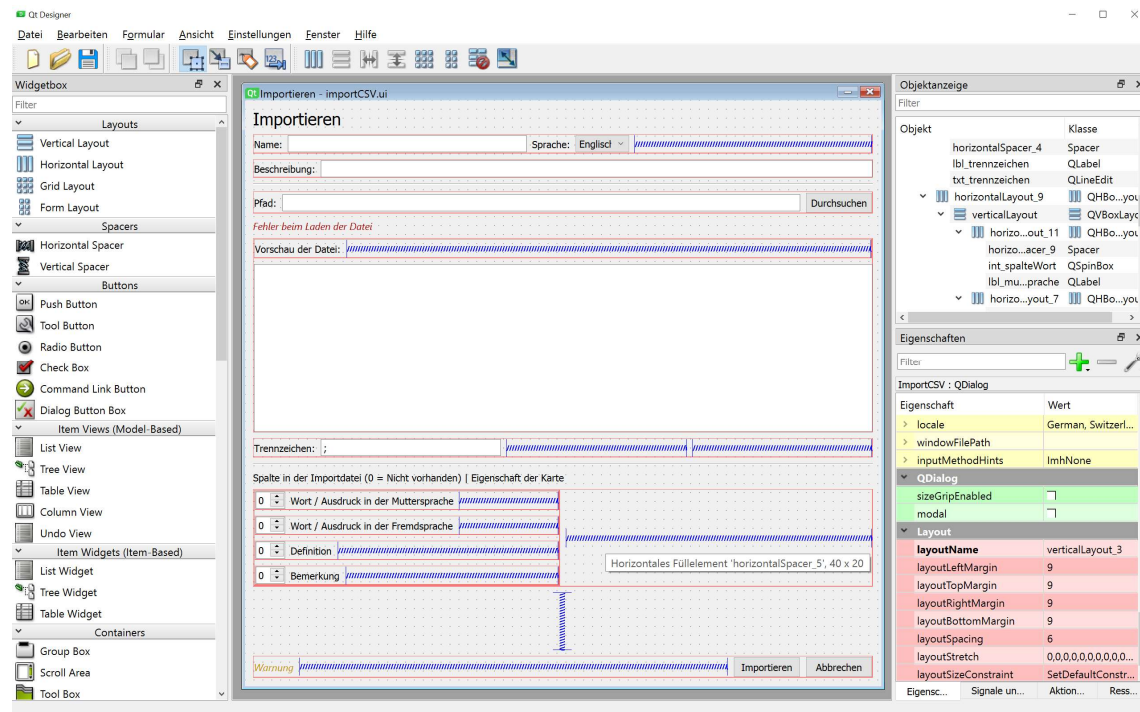
## Objektorientierte Programmierung

Objektorientierte Programmierung kurz OOP wurde mit dem Aufkommen von grafischen Benutzeroberflächen erfunden. Bei den Unmengen an Objekten, Variablen und Funktionen braucht es eine Struktur, welche das Ganze organisiert. OOP heisst kurz gesagt, dass es Objekte, zum Beispiel Klassen gibt, denen man Funktionen und Variablen zuordnen kann, welche man dann Methoden und Attribute nennt. So hat beispielsweise jedes QLabel die Methode QLabel.setText(). Auch Fenster lassen sich am besten in selbst geschriebenen Klassen zusammenfassen.

## QtDesigner

Das zweite Hilfsmittel erleichtert das Leben sehr. In unserem Beispiel zuvor haben wir das Label und den Button selber erstellt und platziert. Das ist bei komplexeren Fenstern kaum mehr möglich. Man müsste immer ausprobieren wie das Fenster aussieht und dann wieder verbessern.

Die Lösung ist der mit Qt mitgelieferte QtDesigner. Er ermöglicht es, dass Fenster per Drag and Drop zusammen zu bauen.



Hier ein Bild des Fensters zum Importieren aus CSV-Dateien. Der wichtigste Bereich ist das Fenster in der Mitte. Links sind die verfügbaren Widgets und rechts unten kann man die Eigenschaften des angewählten Widgets bearbeiten.

Falls Sie selbst gerne mit Python programmieren und interessiert sind, kann ich die Webseite [www.pythonguis.com](http://www.pythonguis.com) für den Einstieg sehr empfehlen.



## Ausblick

---

x

# Anhang

---

## Schlusswort

Diese Arbeit neigt sich dem Ende zu, deshalb möchte ich einen Rückblick wagen. Vor einem Jahr nahm in mir die Idee Gestalt an, einen Vokabeltrainer zu programmieren. Ich wusste damals kaum was über die Programmierung komplexer grafischer Benutzeroberflächen mit Python. Seitdem hat sich vieles getan. In der sich rasch entwickelnden Welt der Informatik ist Lernbereitschaft eine wichtige Eigenschaft. Ich habe viel gelernt und es hat mir viel Spass gemacht. Es ist immer eine Freude, wenn man das neu Gelernte gleich einsetzen kann und sieht, dass es funktioniert. So ist der Vocitrainer entstanden, den hoffentlich noch dem einen oder anderen Schüler eine Hilfe sein wird.

Ich möchte mich bei allen bedanken, die mich auf meinem Abenteuer, den das war es, begleitet haben. Besonderen Dank geht an Herrn Kubba, der mich betreut hat. Er hat mir immer geduldig zugehört, während ich mit Begeisterung meine Fortschritte präsentierte. Aber auch in den Situationen, als ich irgendwo festhing, war Verlass auf Herrn Kubba. Mit seiner Erfahrung konnte er mir immer die entscheidenden Tipps geben.

Besonderen Dank geht auch an meine Eltern und Geschwister. Mit viel Geduld haben sie mir zugehört und viel Verständnis aufgebracht, wenn ich am Abend manchmal in mein Zimmer verschwand und für nichts mehr zu haben war: «Maturaarbeit!»

Ein letzter Dank auch an den Leser, der sich selbst noch für das Schlusswort bemüht. Ich hoffe, die Lektüre hat sich gelohnt.

Fragen zum Vocitrainer sind stets willkommen. Auch wenn jemand das Interesse hat, mit mir denn Vocitrainer weiterzuentwickeln, zögere nicht!

Ich bin per Mail erreichbar: [samuel@barmet.ch](mailto:samuel@barmet.ch)

Code Verfügbar unter

## Quellen

3 : o – Stift siegt gegen Tastatur (Dr. Martin Tomasik)

<https://www.psychologie.uzh.ch/de/bereiche/dev/lifespan/erleben/berichte/handschrift.html>  
(5.12.2023)

Die Gedächtnismodelle (Anna Westhoff)

<https://www.brain-effect.com/magazin/dr-karsten-das-kurzzeitgedaechnis> (14.06.2010)

Halten Sie sich kurz! Mit kurzen Lerneinheiten zum Erfolg (Katharina)

<https://blog.brain-friendly.de/2013/04/halten-sie-sich-kurz-mit-kurzen-lerneinheiten-zum-erfolg/> (19.04.2013)

Lernpausen sind gut fürs Gedächtnis (Max-Planck-Gesellschaft)

<https://www.mpg.de/17299970/0728-psy-lernpausen-sind-gut-fuers-gedaechtnis-155111-x>  
(28.07.2021)

Miller's Law: Die wichtigste Regel im UX-Design (Jessy Kösterke)

<https://t3n.de/news/millers-law-wichtigste-regel-ux-1011492/> (14.04.2018)

Pomodoro-Technik: Wie Sie Ablenkungen widerstehen und Ihre Produktivität erhöhen (IONOS)

<https://www.ionos.de/startupguide/produktivitaet/pomodoro-technik/> (22.05.2023)

Produktiver Arbeiten mit der Pomodoro-Technik (Juliane Felder)

<https://www.fhnw.ch/plattformen/zentrumschreiben/2022/02/18/produktiver-arbeiten-mit-der-pomodoro-technik/> (05.12.2023)

Python 3 – Das umfassende Handbuch (Johannes Ernesti, Peter Kaiser)

Rheinwerk Verlag, Bonn, 7., aktualisierte Auflage 2023

PyQt5 Tutorial – Getting started with PyQt5 (Martin Fitzpatrick):

<https://www.pythonguis.com/tutorials/creating-your-first-pyqt-window/> (13.09.2023)

PyQt5 Python 3 Tutorial (TechWithTim)

<https://www.youtube.com/playlist?list=PLzMbGfZo4-lB8MZfHPLTEHOgzJDDLpYj> (24.4.2020)

Qt Bibliothek (Wikipedia)

[https://de.wikipedia.org/wiki/Qt\\_\(Bibliothek\)](https://de.wikipedia.org/wiki/Qt_(Bibliothek)) (23.09.2023)

Spaced Repetition (Wikipedia)

[https://de.wikipedia.org/wiki/Spaced\\_repetition](https://de.wikipedia.org/wiki/Spaced_repetition) (05.12.2023)

The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information (George A. Miller) (März 1956)

Verfügbar unter:

<https://labs.la.utexas.edu/gilden/files/2016/04/MagicNumberSeven-Miller1956.pdf>

(Bild) The Pomodore Technique Book Cover

<https://www.orellfuessli.ch/shop/home/artikeldetails/A1043641537> (26.12.2023)

Vergessenskurve (Wikipedia)

<https://de.wikipedia.org/wiki/Vergessenskurve> (18.02.2022)

## Hilfsmittel und Bibliotheken

DB Browser (SQLite): <https://sqlitebrowser.org/>

GitHub: <https://github.com/>

Google Fonts: <https://fonts.google.com/>

PyCharm: <https://www.jetbrains.com/de-de/pycharm/>

Python: <https://www.python.org/>

Pythons Standardbibliothek: Socket, pickle, ssl, sqlite, csv, typing, dataclasses

PyQt5: <https://pypi.org/project/PyQt5/>, <https://www.riverbankcomputing.com/software/pyqt/>

Rich: <https://rich.readthedocs.io/en/stable/>

**Kapitel Funktionen (Mit viel Screenshots)**

**Ausblick Kapitel**

---