

Projekt Software

---

# QUINCEQR: EIN QR CODE GENERATOR IN PYTHON

---

Alexander Klee

2023



# Projekt Software

---

## QUINCEQR: EIN QR CODE GENERATOR IN PYTHON

---

Alexander Klee

2023

### Inhaltsverzeichnis

<b>1</b>	<b>Kurze Erläuterung</b>	<b>1</b>
<b>2</b>	<b>Ziele</b>	<b>1</b>
2.1	Optionale Ziele . . . . .	1
2.1.1	GUI . . . . .	1
2.1.2	Logos auf Qr-Codes . . . . .	2

# 1 Kurze Erläuterung

Obwohl QR-Codes (Quick Response Codes) ursprünglich für industrielle Zwecke entwickelt wurden, haben sie schnell vielfältige Anwendungen in nicht-technischen Bereichen gefunden. Der Hauptzweck der QR-Codes bestand darin, als Alternative zu Barcodes mit hoher Datenkapazität zu dienen. Die Integration von QR-Code-Scannern in moderne Mobiltelefone hat erheblich dazu beigetragen, eine breite Benutzerbasis aufzubauen. Aufgrund ihrer einfachen Zugänglichkeit finden QR-Codes heute vielfältige Einsatzmöglichkeiten. Beliebte Anwendungsbereiche umfassen, sind jedoch nicht beschränkt auf, das Öffnen von Weblinks im Browser, das automatische Verbinden mit WLAN-Netzwerken, das Weitergeben von Telefonnummern und anderen Kontaktinformationen, sowie für digitale Zahlungen.

Während die meisten Barcodes lediglich einige Dutzend Buchstaben speichern können, ist die größte QR-Code-Version in der Lage, nahezu 3000 Buchstaben zu speichern. Ebenfalls ein wichtiger Teil des QR-Codes ist die Fehlerkorrektur. Je nach Einstellungen kann bis zu 30% des QR-Codes beschädigt oder verdeckt sein, ohne dass Datenverlust auftritt.

## 2 Ziele

Das Hauptziel dieses Projekts war es, einen funktionsfähigen QR-Code-Generator in Python zu entwickeln, dabei den Einsatz von bereits bestehenden Bibliotheken auf ein Minimum zu beschränken. Das Programm, das sich lediglich auf die Bibliotheken `Numpy` und `Pillow` stützt, erzeugt zuverlässig funktionierende QR-Codes, womit die gesetzten Ziele erreicht wurden. Selbst bei komplexeren mathematischen Operationen wurde auf den Einsatz externer Programme verzichtet. Die für die Reed-Solomon-Codes Fehlerkorrektur benötigte Polynomdivision sowie Rechenoperationen im Galois-Feld-256 wurden vollständig eigenständig implementiert.

### 2.1 Optionale Ziele

#### 2.1.1 GUI

Da das Programm als Python Modul konzipiert wurde eignet es sich dazu in anderen Programmen benutzt zu werden. Nach dem importieren des Moduls kann sehr unkompliziert ein QR-Code erzeugt werden. Das oberste Ziel beim entwickeln der GUI lautete Einfachheit. Nicht nur sollte die Nutzoberfläche keinerlei Untermenüs beinhalten, sondern vor allem sollte die GUI innerhalb eines Tages fertiggestellt werden. Der Fokus dieses Projektes ist schließlich nicht das Design einer guten Benutzerschnittstelle, sondern die GUI dient lediglich Demonstrationszwecken.

Dennoch beinhaltet die GUI einige features:

- Wiederholtes ändern des QR-Code Textes
- Einfaches auswählen von verschiedenen Optionen, wie zum Beispiel dem Grad der Fehlerkorrektur oder der minimalen QR-Code Größe
- Speichern von den generierten QR-Codes

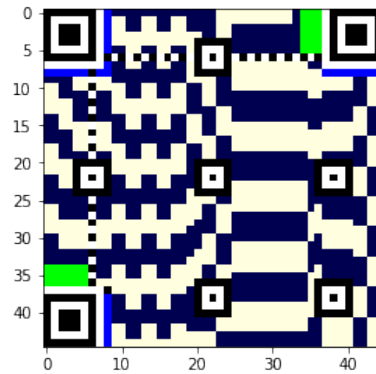


Abbildung 1: Der Aufbau eines QR-Codes der Version 7 mit 30% Fehlerkorrektur. Die einzelnen Reed-Solomon Codewörter sind abwechselnd dunkelblau und beige gekennzeichnet.

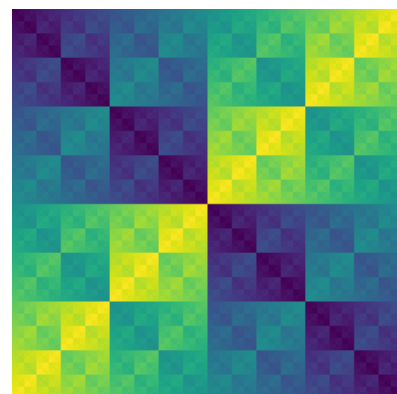


Abbildung 2: Visualisierung von Addition im Galois-Feld 256. Die X- und Y-Achsen haben die Werte 0 bis 255, die Farbe zeigt den Wert des Ergebnisses.

- Einfügen von Logos und anderen Bildern auf den QR-Codes.

### **2.1.2 Logos auf Qr-Codes**

Häufig werden Logos oder Bilder zu QR-Codes hinzugefügt, um dem Benutzer eine intuitivere Erkennung der Funktion des QR-Codes zu ermöglichen. Um solche Logos in QR-Codes einzubetten, wird die Fehlerkorrektur auf ein hohes Niveau (30%) eingestellt und das Bild direkt in den QR-Code eingefügt. Solange das Logo klein genug ist, gehen keine Daten verloren - die Fehlerkorrektur gleicht die resultierende Abdeckung aus.

## Quellen

- [1] Thonky.com. *QR Code Tutorial*. 2022. URL: <https://www.thonky.com/qr-code-tutorial/> (besucht am 16.05.2023).
- [2] *Information technology — Automatic identification and data capture techniques — Bar code symbology — QR Code*. International Organization for Standardization. 2000. URL: [https://www.swisseduc.ch/informatik/theoretische\\_informatik/qr\\_codes/docs/qr\\_standard.pdf](https://www.swisseduc.ch/informatik/theoretische_informatik/qr_codes/docs/qr_standard.pdf) (besucht am 16.05.2023).
- [3] Wikipedia contributors. *QR code — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=QR\\_code&oldid=1154318338](https://en.wikipedia.org/w/index.php?title=QR_code&oldid=1154318338). [Online; accessed 16-May-2023]. 2023.