

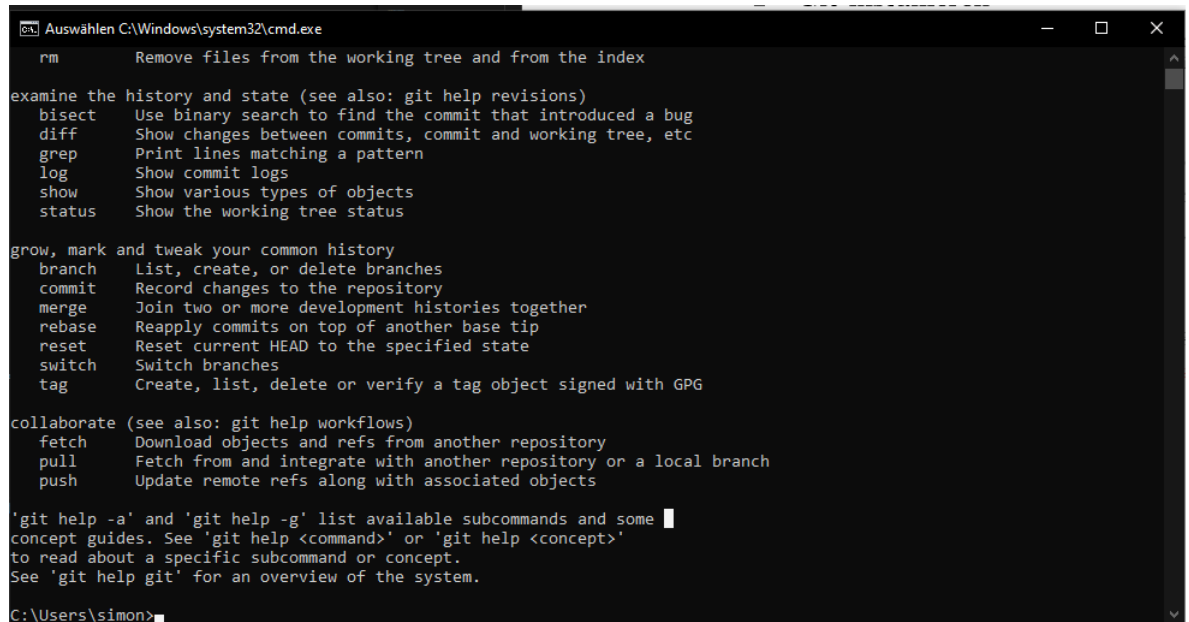
Anleitung

1 Editor installieren

- **Editor:** Dient zum ansehen und bearbeiten von Code
- **Empfohlen:** VSCode mit entsprechenden Erweiterungen
- **Installation:** <https://code.visualstudio.com/>
- **Erweiterungen:**
 - Python
 - Rainbow CSV
 - Better Jinja

2 Git installieren

- **Git:** Programm zur Versionsverwaltung, erlaubt es, Code zu downloaden, und Änderungen vorzunehmen. Zudem kann hier bei Fehlern zu älteren Versionen zurückgesprungen werden
- **Installation:** <https://git-scm.com/download/>
- alle vorgeschlagenen Optionen beibehalten (bis auf Editor)
- Standard Editor: VSCode aus dem Auswahlmenü wählen
- **Test:**
 1. Konsole öffnen (Windows: WinTaste + R ⇒ cmd ⇒ Enter)
 2. git
 3. Folgender Dialog sollte so (oder ähnlich) erscheinen:



```

Auswählen C:\Windows\system32\cmd.exe
rm      Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect  Use binary search to find the commit that introduced a bug
  diff    Show changes between commits, commit and working tree, etc
  grep    Print lines matching a pattern
  log     Show commit logs
  show    Show various types of objects
  status  Show the working tree status

grow, mark and tweak your common history
  branch  List, create, or delete branches
  commit  Record changes to the repository
  merge   Join two or more development histories together
  rebase  Reapply commits on top of another base tip
  reset   Reset current HEAD to the specified state
  switch  Switch branches
  tag     Create, list, delete or verify a tag object signed with GPG

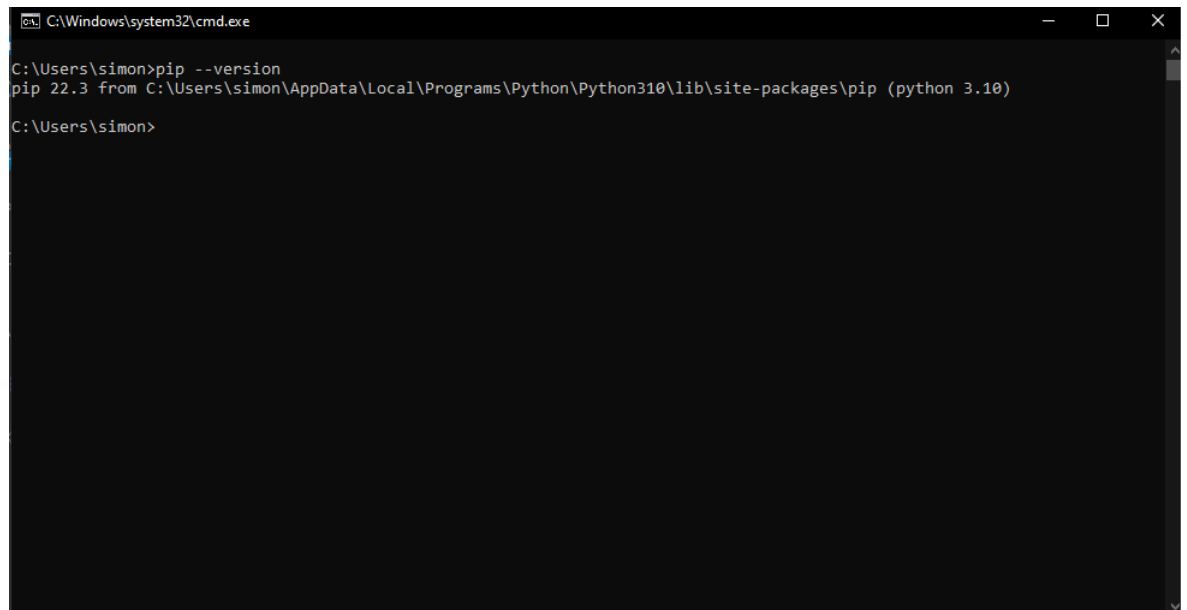
collaborate (see also: git help workflows)
  fetch   Download objects and refs from another repository
  pull    Fetch from and integrate with another repository or a local branch
  push    Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\simon>
```

3 Python installieren

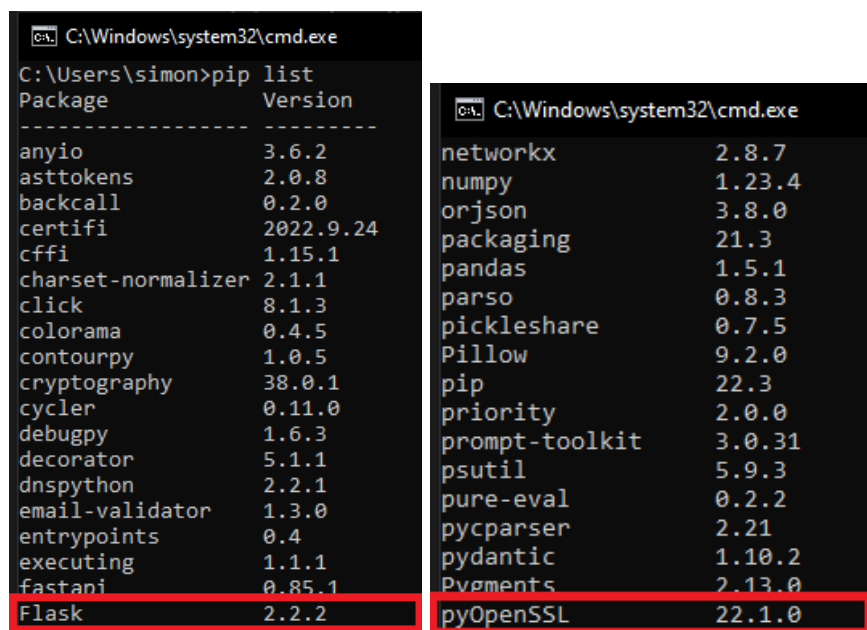
- **Python:** Programmiersprache, in der die App geschrieben ist
- **Installation:** <https://www.python.org/downloads/> Version 3.8 oder höher
- **Test:**
 1. Konsole öffnen (Windows: WinTaste + R \Rightarrow cmd \Rightarrow Enter)
 2. pip --version
 3. Folgender Dialog sollte so (oder ähnlich) erscheinen:



```
C:\Windows\system32\cmd.exe
C:\Users\simon>pip --version
pip 22.3 from C:\Users\simon\AppData\Local\Programs\Python\Python310\lib\site-packages\pip (python 3.10)
C:\Users\simon>
```

4 Pakete installieren

- **Paket:** Code, welcher von anderen Personen wiederverwendet werden kann (z.B. Ansammlung mathematischer Funktionen, welche die Programmiersprache selber nicht direkt bereitstellt)
- **Installation:** Mit Paketmanager **pip**
 1. Konsole öffnen (Windows: WinTaste + R \Rightarrow cmd \Rightarrow Enter)
 2. pip install [<paketname>](#)
- **Pakete:**
 - [flask](#) (Webserver)
 - [pyopenssl](#) (für HTTPS-Verbindungen)
- **Test:**
 - Konsole öffnen (Windows: WinTaste + R \Rightarrow cmd \Rightarrow Enter)
 - pip list
 - folgende Pakete sollten (unter anderem) aufgelistet werden:



```
C:\Windows\system32\cmd.exe
C:\Users\simon>pip list
Package            Version
-----
anyio               3.6.2
asttokens           2.0.8
backcall            0.2.0
certifi             2022.9.24
cffi                1.15.1
charset-normalizer  2.1.1
click               8.1.3
colorama            0.4.5
contourpy           1.0.5
cryptography        38.0.1
cyclor              0.11.0
debugpy             1.6.3
decorator           5.1.1
dnspython           2.2.1
email-validator     1.3.0
entrypoints         0.4
executing           1.1.1
fastapi             0.85.1
Flask               2.2.2
networkx            2.8.7
numpy               1.23.4
orjson              3.8.0
packaging           21.3
pandas              1.5.1
parso               0.8.3
pickleshare         0.7.5
Pillow              9.2.0
pip                 22.3
priority            2.0.0
prompt-toolkit      3.0.31
psutil              5.9.3
pure-eval           0.2.2
pycparser           2.21
pydantic            1.10.2
Pygments            2.13.0
pyOpenSSL           22.1.0
```

5 Projekt downloaden

- Ordner erstellen, in dem das Projekt gespeichert werden soll
- VSCode (Editor) öffnen
- Unter File ⇒ Open Folder den entsprechenden Ordner auswählen
- Unter Terminal ⇒ New Terminal eine Konsole öffnen
- **git clone <https://github.com/BreadMakesUFat/BarcodeScannerDemo>**

6 Setup

- vertrauliche Dateien (config) oder Bibliotheken sind nicht im Projekt enthalten
- **Bibliotheken:**
 - Bootstrap 5: <https://getbootstrap.com/docs/5.0/getting-started/download/>
 - Html5-QRCode: <https://github.com/mebjas/html5-qrcode>
- entsprechende Bibliotheken downloaden und in den Ordner static verschieben
- Referenzen in template anpassen
- **Config:**
 1. neue Datei **config.py** erstellen
 2. folgenden Code übertragen

```
config.py > ...
1  # Development Mode
2  class DevelopmentConfig(object):
3      DEBUG = True
4      DEVELOPMENT = True
5      SCANNER_BOX = (550, 150)
6      HOST =
7      PORT = 8000
8
9
10 # Production mode
11 class ProductionConfig(object):
12     DEBUG = False
13     DEVELOPMENT = False
14     SCANNER_BOX = (550, 150)
15     HOST =
16     PORT = 8000
```

3. IP-Adresse in Anführungszeichen jeweils für HOST eintragen
4. In der Datei **app.py** zu ProductionConfig wechseln:

```
app.py > ...
1  # TODO: improve design
2  # TODO: proper error handling + messages
3  # TODO: improve code structure
4  # TODO: github?
5  # TODO: browser settings to directly access page (add exceptions)
6  # TODO: add updater for git?
7
8  from flask import Flask, render_template, request
9  import csv
10 import datetime
11 import os
12
13 app = Flask(__name__)
14
15 # load config "config.ProductionConfig"
16 app.config.from_object("config.DevelopmentConfig")
17
```

7 Netzwerk einrichten

- Alle Geräte müssen im selben Netzwerk wie das Gerät sein, auf dem der Server läuft
- Lösung: z.B. Handy oder Laptop als Wi-Fi hotspot einrichten und alle Geräte, auf welchen die App laufen soll, damit verbinden

8 App starten

- **Server starten:**

1. VSCode öffnen
2. Ordner öffnen, in dem das Projekt gespeichert ist
3. Konsole in VSCode öffnen
4. **python app.py**

- **App öffnen:**

1. Internetbrowser auf Handy/Barcodescanner öffnen
2. **https://<IP-Adresse>:Port**
Beispiel: **https://127.0.0.1:8000**
3. Unsichere Verbindung akzeptieren
4. Berechtigungen für Kamera geben
5. Seite auf Startbildschirm speichern (optional)