

Klar! Hier kommt dein ausführlicher, gut kommentierter Lernzettel zum Python-Modul **smtplib** – zum **E-Mails versenden** per SMTP, Schritt für Schritt erklärt und praxisnah.

Lernzettel: Python-Modul **smtplib**

1. Was ist **smtplib**?

- **smtplib** ist das Standardmodul, um mit Python **E-Mails über SMTP-Server zu verschicken**.
 - SMTP = Simple Mail Transfer Protocol – das Standardprotokoll für E-Mail-Versand.
 - Du kannst damit dich bei einem SMTP-Server anmelden, E-Mails senden und die Verbindung schließen.
-

2. Grundprinzip: SMTP-Verbindung aufbauen

- Zuerst verbindest du dich mit dem SMTP-Server (z.B. von Gmail, deinem Mailprovider).
 - Dann meldest du dich an (Login), falls nötig.
 - Danach sendest du die E-Mail.
 - Am Ende schließt du die Verbindung.
-

3. Einfaches Beispiel: E-Mail senden

```
import smtplib

# SMTP-Server-Daten
smtp_server = "smtp.example.com"
smtp_port = 587 # 587 für TLS, 465 für SSL

# Zugangsdaten
username = "deine_email@example.com"
password = "dein_passwort"

# Nachricht (einfacher Text)
message = """\
From: deine_email@example.com
To: empfaenger@example.com
Subject: Test E-Mail

Hallo,
das ist eine Test-E-Mail von Python!
"""

# Verbindung aufbauen
server = smtplib.SMTP(smtp_server, smtp_port)
```

```

server.starttls() # TLS-Verschlüsselung starten
server.login(username, password) # Login

# Mail senden
server.sendmail(username, "empfaenger@example.com", message)

# Verbindung beenden
server.quit()

```

Erklärung:

- `SMTP()` baut die Verbindung zum Server auf.
 - `starttls()` verschlüsselt die Verbindung (wichtig!).
 - `login()` authentifiziert dich.
 - `sendmail()` sendet die Mail: Absender, Empfänger, Nachricht als String.
 - `quit()` beendet die Sitzung sauber.
-

4. Wichtige Parameter von `sendmail()`

```
sendmail(from_addr, to_addrs, msg, mail_options=(), rcpt_options=())
```

- `from_addr`: Absenderadresse
 - `to_addrs`: Empfänger, kann eine einzelne Adresse oder Liste sein
 - `msg`: komplette E-Mail als String (inkl. Header und Body)
 - `mail_options, rcpt_options`: selten gebraucht (für spezielle SMTP-Optionen)
-

5. E-Mail mit `email.message.EmailMessage` schicken

Moderne Mails baust du besser mit `email.message.EmailMessage` und schickst die mit `smtplib`:

```

import smtplib
from email.message import EmailMessage

msg = EmailMessage()
msg['From'] = "deine_email@example.com"
msg['To'] = "empfaenger@example.com"
msg['Subject'] = "Test mit EmailMessage"
msg.set_content("Hallo, das ist eine Mail mit EmailMessage!")

with smtplib.SMTP("smtp.example.com", 587) as server:
    server.starttls()
    server.login("deine_email@example.com", "dein_passwort")
    server.send_message(msg)

```

Erklärung:

- `send_message()` nimmt direkt ein `EmailMessage`-Objekt.

- Die Verbindung wird mit `with` automatisch geschlossen.
-

6. SSL statt TLS

Manche SMTP-Server verlangen SSL (Port 465). Dann nutzt du `smtplib.SMTP_SSL`:

```
import smtplib

with smtplib.SMTP_SSL("smtp.example.com", 465) as server:
    server.login("deine_email@example.com", "dein_passwort")
    server.sendmail("deine_email@example.com", "empfaenger@example.com",
message)
```

7. Mehrere Empfänger senden

```
to_emails = ["user1@example.com", "user2@example.com"]
server.sendmail("deine_email@example.com", to_emails, message)
```

Alle bekommen die Mail.

8. Fehlerbehandlung

Es lohnt sich, die Verbindung mit `try/except` abzusichern:

```
try:
    with smtplib.SMTP("smtp.example.com", 587) as server:
        server.starttls()
        server.login(username, password)
        server.send_message(msg)
except Exception as e:
    print("Fehler beim Senden:", e)
```

9. Zusammenfassung

- `smtplib` ermöglicht E-Mail-Versand über SMTP.
 - Baue Verbindung auf, starte TLS, melde dich an, sende Mail, schließe Verbindung.
 - Nutze `EmailMessage` für professionelle Mails.
 - Achte auf korrekte Serverdaten und Ports.
 - Fehlerbehandlung ist wichtig.
-

Falls du willst, kann ich dir auch Beispielcode für `imaplib` zum E-Mails abrufen machen oder eine Kombination aus Lesen (`imaplib + email`) und Schreiben (`smtplib`) zeigen.

Möchtest du das?