

# Do-it-yourself: Statischer HTTP-Server

Wir verwenden wieder einmal die Programmiersprache Python. Die *Socket-Programmierschnittstelle* haben wir ja schon [hier](#) kennengelernt. Deswegen steigen wir direkt in den Code ein:

```
import socket

# Allgemeine Definitionen:
SERVER_HOST = '0.0.0.0' # d.h. alle Netzwerkschnittstellen des Rechners
SERVER_PORT = 8000 # nur im Beispiel, per Konvention geht http über Port 80

# Wir binden uns an einen Socket
server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_socket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
server_socket.bind((SERVER_HOST, SERVER_PORT))
server_socket.listen(1)
print('Lausche auf Port %s ...' % SERVER_PORT)

# Abarbeiten aller Client-Anfragen in einer Endlosschleife:
while True:
    # hier warten wir auf den ersten Client
    client_connection, client_address = server_socket.accept()

    # da hat einer angebissen, jetzt die Daten des Requests:
    request = client_connection.recv(1024).decode()
    print(request)

    # Und unser HTTP Response dazu:
    response = 'HTTP/1.0 200 OK\n\nHallo Welt'
    client_connection.sendall(response.encode())
    client_connection.close()

# Wenn die Endlosschleife am Ende ist...
server_socket.close()
```

From:

<https://schnipsl.qgelm.de/> - Qgelm

Permanent link:

[https://schnipsl.qgelm.de/doku.php?id=schulung:statischer\\_http\\_server&rev=1638456356](https://schnipsl.qgelm.de/doku.php?id=schulung:statischer_http_server&rev=1638456356)

Last update: 2021/12/02 14:45

