

Woche 8: Netzwerke mit Python

Einleitung

Die folgenden drei Wochen unseres Trainings sind einer spannenden und zunehmend wichtigen Disziplin in der Welt der Programmierung gewidmet: der Netzwerkprogrammierung mit Python.

Denn in unserer zunehmend vernetzten Welt ist das Verständnis, wie Daten über Netzwerke übertragen werden, von zentraler Bedeutung. Ob Sie eine Karriere in der System- und Netzwerkadministration, in der Cybersicherheit oder in der Softwareentwicklung anstreben – diese Fähigkeiten sind unerlässlich in einer Vielzahl von Anwendungsbereichen – von der Entwicklung moderner Webanwendungen über die Sicherstellung der Netzwerksicherheit bis hin zur Automatisierung von Systemadministration.

In dieser Woche beginnen wir mit den Grundlagen. Sie werden die Grundprinzipien der Netzwerkkommunikation kennenlernen, einschließlich der Funktionsweise verschiedener Netzwerkprotokolle wie TCP/IP und UDP. Wir werden untersuchen, wie Python für die Erstellung einfacher Netzanwendungen genutzt werden kann, und Sie werden lernen, wie man grundlegende Client- und Server-Anwendungen in Python erstellt. Diese Woche ist darauf ausgerichtet, Ihnen ein festes Fundament in der Netzwerkprogrammierung zu geben, auf dem die nächsten Wochen aufbauen werden.

In dieser ersten Woche zum Thema Netzwerkprogrammierung mit Python, behandeln wir folgende Themen:

- Einführung in Netzwerkprotokolle und Kommunikationsmodelle.

- Praktische Erfahrungen mit der Erstellung von Client-Server-Applikationen in Python.

- Einblicke in die Handhabung von Netzwerkdaten und die Bearbeitung von Netzwerkfehlern.

- Projekte und Übungen, die Ihr Wissen festigen und erweitern.

Bereiten Sie sich darauf vor, in die Welt der Netzwerkprogrammierung einzutauchen und Fähigkeiten zu entwickeln, die Sie in die Lage versetzen, leistungsfähige und effiziente Netzanwendungen zu erstellen.

Gesamtüberblick

Hier ein Überblick über die Inhalte und Aktivitäten der aktuellen Woche:

- Selbststudium:
 - Einführung in die Netzwerkkommunikation und Verständnis grundlegender Netzwerkprotokolle wie TCP/IP und UDP.
 - Untersuchung des Client-Server-Modells und dessen Funktionsweise.
 - Erstellen und Testen von einfachen Client- und Server-Anwendungen in Python.
 - Verstehen und Handhaben von Netzwerk-Sockets und deren Kommunikationsabläufen.
 - Methoden der Datenübertragung und -empfang über Netzwerke.
 - Verarbeitung und Manipulation von Netzwerkdaten in Python.
- Aufgaben:
 - Entwicklung einer einfachen Client-Server-Anwendung, die Basisfunktionen wie das Senden und Empfangen von Nachrichten umfasst.
 - Experimentieren mit verschiedenen Arten von Netzwerkkommunikation und dem Umgang mit Netzwerkfehlern

Inhalte und Thematisch Abgrenzung

Die folgende Auflistung zeigt detailliert, welche Themen Sie in der Woche behandeln und bearbeiten. Sie sind eine Voraussetzung für die folgenden Wochen und sollten gut verstanden worden sein. Wenn es Verständnisprobleme gibt, machen Sie sich Notizen und fragen Sie am Präsenztage nach, so dass wir gemeinsam zu Lösungen kommen können. Und denken Sie bitte immer daran: es gibt keine „dummen“ Fragen!

1. Einführung in die Netzwerkprogrammierung:
 - Grundlagen und Konzepte der Netzwerkprogrammierung.
 - Verständnis von Netzwerkprotokollen wie TCP/IP und UDP.
2. Erstellen von Client- und Server-Anwendungen in Python:
 - Grundlagen der Socket-Programmierung.
 - Erstellen einfacher Client- und Server-Anwendungen mit Sockets in Python.
3. Datenübertragung über Netzwerke:
 - Methoden zum Senden und Empfangen von Daten über Netzwerke.
 - Handhabung von Datenströmen und Nachrichten in Netzerkanwendungen.
4. Netzwerkfehler und Ausnahmen:
 - Erkennen und Behandeln von Netzwerkfehlern in Python.
 - Implementierung robuster Fehlerbehandlungsmechanismen.
5. Aspekte der Netzwerksicherheit:
 - Einführung in die grundlegenden Konzepte der Netzwerksicherheit.
 - Verwendung sicherer Protokolle und Techniken in der Netzwerkprogrammierung.
6. Multithreading in Netzerkanwendungen:
 - Einführung in Multithreading und dessen Anwendung in der Netzwerkprogrammierung.
 - Erstellen von Serveranwendungen, die mehrere Client-Verbindungen gleichzeitig verwalten können.
7. Fortgeschrittene Netzwerkprogrammierung:
 - Vertiefung in fortgeschrittenere Themen wie asynchrone Programmierung.
 - Entwicklung komplexerer Netzerkanwendungen unter Verwendung fortgeschrittener Techniken.
8. Projektarbeit und praktische Anwendung:
 - Anwendung der gelernten Konzepte in einem umfangreichen Netzwerkprojekt.
 - Entwicklung, Testen und Optimierung eigener Netzerkanwendungen.

Lernpfad

Der Lernpfad ist ein Vorschlag, in welcher Reihenfolge Sie die Inhalte der Woche angehen können. Betrachten Sie ihn gerne als eine Todo-Liste, die Sie von oben nach unten abhaken. So können Sie sicher sein, dass Sie alle wichtigen Themen bearbeitet haben und sind gut vorbereitet für die folgenden Wochen.

1. Einführung in die Netzworkkommunikation:
 - Beginnen Sie mit den Grundlagen der Netzworkkommunikation und den zentralen Netzwerkprotokollen wie TCP/IP und UDP.
 - Untersuchen Sie die Funktionen und Eigenschaften des Client-Server-Modells.
2. Erstellen einfacher Netzworkeanwendungen:
 - Lernen Sie, wie man grundlegende Client- und Server-Anwendungen in Python erstellt.
 - Experimentieren Sie mit der Erstellung und dem Testen von Netzwerk-Sockets.
3. Datenübertragung und -verarbeitung:
 - Erforschen Sie Methoden zur Übertragung von Daten über Sockets und deren Verarbeitung in Python.
 - Üben Sie das Senden und Empfangen von Nachrichten zwischen einem Client und einem Server.
4. Netzwerkprotokolle in der Praxis:
 - Vertiefen Sie Ihr Verständnis für TCP und UDP durch praktische Anwendungsfälle.
 - Erstellen Sie einfache Anwendungen, die diese Protokolle nutzen.
5. Fehlerbehandlung in der Netzworkeprogrammierung:
 - Lernen Sie, wie Sie mit Netzworkefehlern und Ausnahmen umgehen.
 - Üben Sie die Implementierung robuster Fehlerbehandlungsmechanismen in Ihren Netzworkeanwendungen.
6. Projektaufgabe: Client-Server-Kommunikation:
 - Entwickeln Sie ein Projekt, das eine Client-Server-Kommunikation umfasst, wie z.B. einen einfachen Nachrichtenaustausch oder Datentransfer.
 - Testen und optimieren Sie Ihre Anwendung für Effizienz und Zuverlässigkeit.

Programmieraufgaben

Die folgenden Programmieraufgaben sollen Ihnen eine Anregung geben. Haben Sie eigene Ideen und Themen, die Sie ausprobieren wollen, dann sollten Sie diesen nachgehen. Wichtig ist vor allem, dass Sie „Dinge ausprobieren“. Und auch, dass Sie Fehler machen, sowohl syntaktische als auch semantische. Versuchen Sie diese Fehler zu finden und aufzulösen, dann gerade aus den Fehlern lernen Sie am Ende am meisten.

1. Erstellen eines einfachen Echo-Servers:

Entwickeln Sie einen einfachen Server, der empfangene Nachrichten an den Client zurücksendet (Echo). Verwenden Sie Sockets in Python, um einen Server zu erstellen, der Verbindungen von Clients akzeptiert und die empfangenen Nachrichten zurücksendet.

2. Entwicklung eines Chat-Clients:

Erstellen Sie einen einfachen Chat-Client, der es ermöglicht, Nachrichten an einen Server zu senden und Antworten zu empfangen. Implementieren Sie einen Client, der sich mit Ihrem Echo-Server verbindet und eine bidirektionale Kommunikation ermöglicht.

3. Dateiübertragung zwischen Client und Server:

Entwickeln Sie eine Anwendung, die das Hoch- und Herunterladen von Dateien zwischen einem Client und einem Server ermöglicht. Implementieren Sie sowohl Server- als auch Client-Seite, die es erlauben, Dateien in beide Richtungen zu übertragen.

4. Implementierung eines einfachen HTTP-Servers:

Erstellen Sie einen einfachen HTTP-Server, der grundlegende Webanfragen beantworten kann. Nutzen Sie die Python-Standardbibliothek, um einen Server zu bauen, der einfache HTTP-Anfragen verarbeiten und HTML-Inhalte zurücksenden kann.

Ressourcen

Hier nun die Verweise auf Lernquellen, die uns für diese Woche und ihre Inhalte geeignet erscheinen. Je nachdem, welcher Lerntyp Sie sind, wählen Sie sich ihre bevorzugte Quelle, es ist nicht zwingend notwendig alle durchgearbeitet zu haben. Allerdings sollten die Inhalte des Lernpfads angesprochen und verstanden worden sein.

- **Buch:** Introduction to Python Network Automation
 - Paramiko und Netmiko
 - CISCO Router
- **Buch:** Network Programming in Python: The Basics
 - Umfangreiche Einführung in LowLevel Funktionalität im Netz
 - Alle wichtigen Protokolle
- **Videos:**
 - Python Web Requests
 - Building Restful Web Services with Python
- **Lab and Course:**
 - Lab: Building Restful Web Services with Python
 - Course: Learn Web Scraping with Beautiful Soup