Python für Netzwerker

Praktische Anwendung im Netzwerk

Der Umgang mit Netzwerken zeigt einen klaren Trend, der das klassische CLI als Zugang ablöst oder ergänzt um moderne Application Programming Interfaces (APIs). Diese erstrecken sich von APIs für Router und Switches über APIs für SDN-Controller bis hin zu APIs für Cloud-Infrastruktur. Dieser Kurs erläutert, wie die unterschiedlichen, gängigen APIs in Python-Programmen verwendet werden können.

Kursinhalt

- ipaddress: IPv4- und IPv6-Adressen mit Python verarbeiten
- pysnmp: ein SNMP-Paket für Python
- netmiko: CLI-Befehle in Python-Skripte einbetten
- json: JSON-Dokumente parsen oder erzeugen
- PyYaml und ruamel.yaml: YAML-Dokumente parsen oder erzeugen
- lxml: XML-Dokumente parsen oder erzeugen
- requests: HTTPs-Anfragen z.B. an REST APIs stellen oder RESTCONF-Anfragen formulieren
- ncclient: Client-seitige Python Library für NETCONF
- pyang: YANG-Modellen visualisieren und konvertieren
- pyangbind: Plug-In für pyang zum Parsen von YANG-Modellen und -Instanzen

Zielgruppe

Der Kurs richtet sich an Netzwerkplaner und -administratoren, die auf Grund der sich zunehmend verändernden Anforderungen einen schnellen Einstieg in den Umgang mit modernen APIs benötigen.

Voraussetzungen

Grundkenntnisse der Programmierung in Python sind erforderlich. Der Kurs Python für Einsteiger – Einführung in die Programmierung hilft Ihnen diese Kenntnisse zu erlangen.

Kursziel

Nach diesem Kurs kennen Sie wichtige Module und Pakete für die Programmiersprache Python, die man in der Netzwerkautomatisierung einsetzen kann. Sie können Skripte verfassen, welche diese Module und Pakete verwenden, um z. B. Router oder Switches zu konfigurieren oder auszulesen. Auch die REST API von SDN-Controllern können sie mithilfe von Python-Skripten ansprechen.

Dieser Kurs im Web

Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link: www.experteach.ch/go/**PRPN**

Vormerkung

Sie können auf unserer Website einen Platz kostenlos und unverbindlich für 7 Tage reservieren. Dies geht auch telefonisch unter 06074 4868-0.

Garantierte Kurstermine

Für Ihre Planungssicherheit bieten wir stets eine große Auswahl garantierter Kurstermine an.

Ihr Kurs maßgeschneidert

Diesen Kurs können wir für Ihr Projekt exakt an Ihre Anforderungen anpassen.

Premium Print Paket

Zu diesem Kurs können sie optional das hochwertige Premium Print Paket zum Preis von € 195,- (zzgl. MwSt) erwerben.

Training	Preise zzgl. MwSt.
Termine in Deutschland	**4 Tage CHF 2.635,-**
Termine in Österreich	**4 Tage CHF 2.635,-**
Online Training	**4 Tage CHF 2.635,-**

Termin/Kursort		Kurssprache Deutsch	
18.01.-21.01.27	Frankfurt	12.07.-15.07.27	Frankfurt
18.01.-21.01.27	Online	12.07.-15.07.27	Online
13.04.-16.04.27	München	18.10.-21.10.27	Online
13.04.-16.04.27	Online	18.10.-21.10.27	Wien

Stand 01.11.2026

Inhaltsverzeichnis

Python für Netzwerker – Praktische Anwendung im Netzwerk

1 Wiederholung der Grundkonzepte der Programmiersprache Python
1.1 Die Programmiersprache Python
1.1.1 Download, Installation und Lieferumfang
1.1.2 Python-Shell
1.2 Kurze Demonstration grundlegender Kontroll- und Datenstrukturen
1.2.1 Rechnen mit ganzen Zahlen
1.2.2 Verwendung von Variablen
1.2.3 Gleitkommazahlen und Zeichenfolgen
1.2.4 Listen und Dictionaries
1.2.5 Logische Ausdrücke
1.2.6 Funktionen
1.2.7 Funktionsdefinitionen und Bedingungen
1.2.8 Schleifen und Klassen
1.3 Kontroll- und Datenstrukturen
1.3.1 Anweisungen und Blöcke in Python
1.3.2 Boolesche und Arithmetische Operationen
1.3.3 Weitere Ausdrücke in Python
1.3.4 Variablen in Python
1.3.5 Funktionen I: Eingebaute Funktionen
1.3.6 Die Hilfe-Funktion
1.3.7 Die Bedingung: If
1.3.8 String Details
1.3.9 Weitere String Details
1.3.10 Iterationen I – While
1.3.11 While-Schleife
1.3.12 For – Eine Vorschau
1.3.13 Eigene Funktionen
1.3.14 Verschachtlung
1.3.15 Global oder Lokal
1.3.16 Rekursion
1.3.17 Arten von Argumenten
1.3.18 Liste
1.3.19 Weitere Methoden von Listen
1.3.20 Tupel
1.3.21 Eine Menge - Set
1.3.22 Dictionaries
1.3.23 Dictionaries – Methoden
1.4 Weitere Konzepte und Funktionen
1.4.1 Exceptions
1.4.2 Anonyme Funktionen
1.4.3 Comprehensions
1.5 Objektorientierte Programmierung
1.5.1 Klassendefinition
1.5.2 Verhalten hinzufügen
1.5.3 Spezialisierung durch Subklassen
1.6 Hinzufügen von internen und externen Modulen in Python
1.6.1 Import
1.6.2 Regular Expressions
1.6.3 Externe Module: Pip und PyPI

2 Grundlagen der Netzwerkkommunikation
2.1 Die Zutaten zu einem IP-Netz
2.2 Software und Netzwerke
2.2.1 Client-/Server-Konzept
2.3 Arbeitsteilung – das Schichtenmodell
2.3.1 Die Anwendungsschicht
2.3.2 Die Transportschicht
2.3.3 Die Netzwerkschicht
2.3.4 Die Teilstreckenschicht
2.3.5 Die Bitübertragung
2.3.6 Alles zusammen
2.3.7 Die Protokollfamilie
2.3.8 Zwischen Anwendung und IP: TCP und UDP
2.3.9 Source und Destination Port
2.3.10 UDP
2.3.11 TCP
2.4 Kommunikationsendpunkte: Sockets
2.4.1 Datagramm oder Bytestrom
2.4.2 TCP
2.4.3 UDP

3 Anwendungen im Netzwerkbereich mit Python
3.1 Das Modul ipaddress
3.1.1 Das Parameter von ip_address()
3.1.2 IPv4Address und IPv6Address
3.1.3 Die Funktion ip_network()
3.2 Kommunikationsendpunkte in Python: Das Sockets-Modul
3.3 Network Management: SNMP und pysnmp
3.3.1 Das Modul pysnmp
3.3.2 Das Modul pysnmp – getCmd
3.4 Konfigurieren klassischer Netzwerk-Devices
3.4.1 Paramiko – SSH-Verbindungen
3.4.2 Netmiko – Konfiguration eines Netzwerkgerätes
3.4.3 Netmiko – Konfiguration mehrerer Netzwerkgeräte
3.4.4 Credentials aus einer Datei einlesen
3.5 Verwendung von APIs
3.5.1 Betriebssystem API
3.5.2 Router API
3.5.3 Netzwerk-API
3.5.4 REST API
3.6 REST in Python – Das requests-Paket
3.6.1 Das response-Objekt
3.6.2 Funktionsparameter
3.7 API-Datenaustauschformate: JSON
3.7.1 JSON in Python
3.8 API-Datenaustauschformate: XML
3.8.1 XML in Python – ElementTree
3.8.2 XML-Parsing
3.9 API-Datenaustauschformate: YAML
3.10 Netzwerkmanagement: NETCONF
3.10.1 Die Funktionsweise von NETCONF
3.10.2 NETCONF in Python
3.10.3 Eine Netconf-Session mit ncclient
3.10.4 Das Manager-Objekt und der Context Handler
3.10.5 Subtree Filter
3.11 Netzwerkmodelle: YANG
3.11.1 Augmentations und Deviations
3.11.2 pyang
3.11.3 Abhängigkeiten als UML-Diagramm
3.12 pyATS – Python Automated Test Systems
3.13 Software Defined Networking
3.13.1 Klassische Router/Switch-Netze
3.13.2 Software Defined Networking
3.13.3 Controller und Protokolle
3.13.4 Virtuelle Netze
3.14 Weitere Anwendungen: OpenStack
3.14.1 Geschichte
3.14.2 Module von OpenStack
3.14.3 SDN und Neutron
3.15 Weitere APIs

ExperTeach AG
Kronenstrasse 11 · 8735 St. Gallenkappel · Telefon: +41 55 420 2591 · Fax: +41 55 420 2592 · info@experteach.ch · www.experteach.ch