

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

- Klassische Router und Switches
- REST
- NETCONF
- RESTCONF
- gRPC
- SDN-Controller

E-Book Das ausführliche deutschsprachige digitale Unterlagenpaket, bestehend aus PDF und E-Book, ist im Kurspreis enthalten.

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.

Dieser Kurs im Web



Alle tagesaktuellen Informationen und Möglichkeiten zur Bestellung finden Sie unter dem folgenden Link: www.experteach.de/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.

Training		Preise zzgl. MwSt.
Termine in Deutschland	4 Tage	€ 2.395,-
Termine in Österreich	4 Tage	€ 2.395,-
Online Training	4 Tage	€ 2.395,-
Termin/Kursort	Kurssprache Deutsch	
10.06.-13.06.25	Online	03.02.-06.02.26 Online
26.08.-29.08.25	Frankfurt	03.02.-06.02.26 Wien
26.08.-29.08.25	Online	07.04.-10.04.26 Frankfurt
21.10.-24.10.25	München	07.04.-10.04.26 Online
21.10.-24.10.25	Online	16.06.-19.06.26 Hamburg
15.12.-18.12.25	Frankfurt	16.06.-19.06.26 Online
15.12.-18.12.25	Online	

Stand 03.06.2025



Inhaltsverzeichnis

Python für Netzwerker – Praktische Anwendung im Netzwerk

1	Wiederholung der Grundkonzepte der Programmiersprache Python	1.5.3	Spezialisierung durch Subklassen	3.5	Verwendung von APIs
1.1	Die Programmiersprache Python	1.6	Hinzufügen von internen und externen Modulen in Python	3.5.1	Betriebssystem API
1.1.1	Download, Installation und Lieferumfang	1.6.1	Import	3.5.2	Router API
1.1.2	Python-Shell	1.6.2	Regular Expressions	3.5.3	Netzwerk-API
1.2	Kurze Demonstration grundlegender Kontroll- und Datenstrukturen	1.6.3	Externe Module: Pip und PyPI	3.5.4	REST API
1.2.1	Rechnen mit ganzen Zahlen	2	Grundlagen der Netzwerkkommunikation	3.6	REST in Python – Das requests-Paket
1.2.2	Verwendung von Variablen	2.1	Die Zutaten zu einem IP-Netz	3.6.1	Das response-Objekt
1.2.3	Gleitkommazahlen und Zeichenfolgen	2.2	Software und Netzwerke	3.6.2	Funktionsparameter
1.2.4	Listen und Dictionaries	2.2.1	Client-/Server-Konzept	3.7	API-Datenaustauschformate: JSON
1.2.5	Logische Ausdrücke	2.3	Arbeitsteilung – das Schichtenmodell	3.7.1	JSON in Python
1.2.6	Funktionen	2.3.1	Die Anwendungsschicht	3.8	API-Datenaustauschformate: XML
1.2.7	Funktionsdefinitionen und Bedingungen	2.3.2	Die Transportschicht	3.8.1	XML in Python – ElementTree
1.2.8	Schleifen und Klassen	2.3.3	Die Netzwerkschicht	3.8.2	XML-Parsing
1.3	Kontroll- und Datenstrukturen	2.3.4	Die Teilstreckenschicht	3.9	API-Datenaustauschformate: YAML
1.3.1	Anweisungen und Blöcke in Python	2.3.5	Die Bitübertragung	3.10	Netzwerkmanagement: NETCONF
1.3.2	Boolesche und Arithmetische Operationen	2.3.6	Alles zusammen	3.10.1	Die Funktionsweise von NETCONF
1.3.3	Weitere Ausdrücke in Python	2.3.7	Die Protokollfamilie	3.10.2	NETCONF in Python
1.3.4	Variablen in Python	2.3.8	Zwischen Anwendung und IP: TCP und UDP	3.10.3	Eine Netconf-Session mit ncclient
1.3.5	Funktionen I: Eingebaute Funktionen	2.3.9	Source und Destination Port	3.10.4	Das Manager-Objekt und der Context Handler
1.3.6	Die Hilfe-Funktion	2.3.10	UDP	3.10.5	Subtree Filter
1.3.7	Die Bedingung: If	2.3.11	TCP	3.11	Netzwerkmodelle: YANG
1.3.8	String Details	2.4	Kommunikationsendpunkte: Sockets	3.11.1	Augmentations und Deviations
1.3.9	Weitere String Details	2.4.1	Datagramm oder Bytestrom	3.11.2	pyang
1.3.10	Iterationen I – While	2.4.2	TCP	3.11.3	Abhängigkeiten als UML-Diagramm
1.3.11	While-Schleife	2.4.3	UDP	3.12	pyATS – Python Automated Test Systems
1.3.12	For – Eine Vorschau	3	Anwendungen im Netzwerkbereich mit Python	3.13	Software Defined Networking
1.3.13	Eigene Funktionen	3.1	Das Modul ipaddress	3.13.1	Klassische Router/Switch-Netze
1.3.14	Verschachtlung	3.1.1	Das Parameter von ip_address()	3.13.2	Software Defined Networking
1.3.15	Global oder Lokal	3.1.2	IPv4Address und IPv6Address	3.13.3	Controller und Protokolle
1.3.16	Rekursion	3.1.3	Die Funktion ip_network()	3.13.4	Virtuelle Netze
1.3.17	Arten von Argumenten	3.2	Kommunikationsendpunkte in Python: Das Sockets-Modul	3.14	Weitere Anwendungen: OpenStack
1.3.18	Liste	3.3	Network Management: SNMP und pysnmp	3.14.1	Geschichte
1.3.19	Weitere Methoden von Listen	3.3.1	Das Modul pysnmp	3.14.2	Module von OpenStack
1.3.20	Tupel	3.3.2	Das Modul pysnmp – getCmd	3.14.3	SDN und Neutron
1.3.21	Eine Menge - Set	3.4	Konfigurieren klassischer Netzwerk-Devices	3.15	Weitere APIs
1.3.22	Dictionaries	3.4.1	Paramiko – SSH-Verbindungen		
1.3.23	Dictionaries – Methoden	3.4.2	Netmiko – Konfiguration eines Netzwerkgerätes		
1.4	Weitere Konzepte und Funktionen	3.4.3	Netmiko – Konfiguration mehrerer Netzwerkgeräte		
1.4.1	Exceptions	3.4.4	Credentials aus einer Datei einlesen		
1.4.2	Anonyme Funktionen				
1.4.3	Comprehensions				
1.5	Objektorientierte Programmierung				
1.5.1	Klassendefinition				
1.5.2	Verhalten hinzufügen				

