

コースコード：CCNAAUTO

税込価格：363,000円 (税抜価格：330,000円)

日数：5日間

ここに注目!!

Cisco Automation認定資格のCCNAAUTO試験（試験番号: 200-901）に対応しています。

受講対象者

このトレーニングはこのような方におすすめです。

- ・ Cisco CCNA Automation認定の取得を目指す方
- ・ ネットワーク自動化エンジニア
- ・ ソフトウェア開発者
- ・ システムインテグレーションプログラマ
- ・ インフラストラクチャアーキテクト
- ・ ネットワーク設計者

前提条件/前提知識

このコースを受講する前に受講者が習得しておく必要がある知識およびスキルは次のとおりです。

- ・ 基本的なコンピュタリテラシー
- ・ 基本的なPC操作スキル
- ・ 基本的なインターネット利用に関するスキル
- ・ プログラミング言語（特にPython）の利用経験

目的

このコースを修了すると次のことができるようになります。

- ・ 現代のソフトウェア開発におけるAPIの重要性とバージョン管理ツールの使用について説明する
- ・ ソフトウェア開発で用いられる一般的なプロセスと実践について説明する
- ・ モジュール化されたソフトウェアを組織化し構築するための選択肢について説明する
- ・ HTTPの概念と、それらがネットワークベースのAPIにどのように適用されるかを説明する
- ・ Representational State Transfer（REST）の概念をHTTPベースのAPIとの統合に応用する
- ・ Ciscoプラットフォームとその機能について説明する
- ・ 異なるCiscoプラットフォームのプログラマビリティ機能について説明する
- ・ 基本的なネットワーク概念を説明し、単純なネットワークポロジを解釈する
- ・ アプリケーションとネットワークの相互作用、および問題解決に用いられるツールを説明する
- ・ モデル駆動型プログラマビリティの概念を適用し、Pythonスクリプトで一般的なタスクを自動化する
- ・ 一般的なアプリケーション展開モデルと開発パイプラインの構成要素を特定する
- ・ スクリプティングとモデル駆動型プログラマビリティを通じてインフラを自動化するツールを活用する
- ・ 一般的なセキュリティ上の懸念事項とテストの種類を説明し、ローカル開発にコンテナ化を活用する



アウトライン

講義アウトライン

- Practicing Modern Software Development
- Describing Software Development Process
- Designing Software
- Introducing Network-Based APIs
- Consuming REST-Based APIs
- Introducing Cisco Platforms and APIs
- Employing Programmability on Cisco Platforms
- Describing IP Networks
- Relating Network and Applications
- Employing Model-Driven Programmability
- Deploying Applications
- Automating Infrastructure
- Testing and Securing Applications
- Lab Code Reference

ラボアウトライン

- Parse API Data Formats with Python
- Use Git for Version Control
- Identify Software Architecture and Design Patterns on a Diagram
- Implement Singleton Pattern and Abstraction-Based Method
- Inspect HTTP Messages
- Use Postman
- Troubleshoot an HTTP Error Response
- Utilize APIs with Python
- Use the Cisco Webex Collaboration API
- Interpret a Basic Network Topology Diagram



Identify the Cause of Application Connectivity Issues

Perform Basic NETCONF Operations

Utilize Bash Commands for Local Development

Construct Infrastructure Automation Workflow

Construct a Python Unit Test

Interpret a Dockerfile

Utilize Docker Commands to Manage Local Developer Environment

Exploit Insufficient Parameter Sanitization