

Automatisierung mit Python

Wiederkehrende Aufgaben mit Python zuverlässig automatisieren

01 INTRO

Viele Aufgaben in Entwicklung, Administration und technischen Prozessen bestehen aus wiederkehrenden Arbeitsschritten: Dateien verarbeiten, Daten konvertieren, Programme aufrufen, Verzeichnisse durchsuchen oder Abläufe miteinander verbinden.

Der Kurs zeigt, wie sich solche Aufgaben mit Python pragmatisch automatisieren lassen. Im Mittelpunkt stehen kleine und mittlere Werkzeuge, die reale Arbeitsabläufe vereinfachen und dabei nachvollziehbar, robust und wartbar bleiben.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Python-Grundkenntnisse

02 TRAININGSPROGRAMM

Inhalte

Automatisierungsaufgaben erkennen und strukturieren

- geeignete Aufgaben für Automatisierung
- wiederkehrende manuelle Abläufe analysieren
- Eingaben, Verarbeitung und Ausgaben trennen
- kleine Skripte oder größere Werkzeuge?
- Fehlerfälle von Anfang an berücksichtigen

Dateien und Verzeichnisse

- Dateien lesen und schreiben
- pathlib
- Verzeichnisse durchsuchen
- Dateien suchen und filtern
- Dateien kopieren, verschieben und umbenennen
- Verzeichnisstrukturen erzeugen
- temporäre Dateien und Verzeichnisse

Text- und Datenverarbeitung

- Textdateien verarbeiten
- CSV-Dateien lesen und schreiben
- JSON verarbeiten
- Daten filtern und transformieren
- reguläre Ausdrücke
- strukturierte Informationen aus Texten extrahieren
- unterschiedliche Datenquellen zusammenführen

Betriebssystem und Umgebung

- Umgebungsvariablen
- Kommandozeilenargumente
- Pfade plattformunabhängig behandeln
- Systeminformationen auswerten
- Dateien und Prozesse überwachen
- Unterschiede zwischen Linux und Windows berücksichtigen

Externe Programme steuern

- Prozesse mit subprocess starten
- Argumente sicher übergeben
- Standardausgabe und Fehlerausgabe auswerten
- Exit Codes prüfen
- Timeouts
- Prozesse kontrolliert abbrechen
- Shell-Aufrufe vermeiden, wenn sie nicht erforderlich sind

Kommandozeilenwerkzeuge entwickeln

- Kommandozeilenparameter
- argparse
- Optionen und Schalter
- Unterkommandos
- Hilfe und Fehlermeldungen
- sinnvolle Exit Codes
- kleine CLI-Werkzeuge strukturieren

Web und HTTP automatisieren

- HTTP-Anfragen aus Python
- REST-Schnittstellen aufrufen
- JSON-Antworten verarbeiten
- Header und Parameter
- Authentifizierung im Überblick
- Fehler und Timeouts behandeln
- Dateien herunterladen und übertragen

Konfiguration und Secrets

- Konfigurationsdateien
- Umgebungsvariablen
- unterschiedliche Laufzeitumgebungen
- Zugangsdaten nicht im Quellcode speichern
- Konfiguration und Programmlogik trennen

Robuste Automatisierung

- Exceptions gezielt behandeln
- fehlende Dateien und Ressourcen
- Netzwerk- und Prozessfehler
- Wiederholungsversuche
- idempotente Abläufe
- Zwischenergebnisse und Wiederanlauf
- kontrolliertes Beenden

Logging und Nachvollziehbarkeit

- logging
- Log-Level
- strukturierte Meldungen
- Logdateien
- Fortschritt und Fehler dokumentieren
- aussagekräftige Diagnoseinformationen
- Logging statt verteilter print-Ausgaben

Zeitgesteuerte und wiederkehrende Abläufe

- Aufgaben regelmäßig ausführen
- Cron und Task Scheduler
- zeitgesteuerte Python-Skripte
- Locking und parallele Starts
- Status und Ergebnis eines Laufs erfassen
- typische Probleme unbeaufsichtigter Prozesse

Automatisierung testen

- Automatisierungslogik testbar strukturieren
- Dateisystemzugriffe testen
- temporäre Ressourcen
- externe Prozesse isolieren
- HTTP-Aufrufe kontrollieren
- kritische Abläufe vor dem produktiven Einsatz prüfen

Vom Skript zum Werkzeug

- Skripte sinnvoll strukturieren
- Funktionen und Module
- Wiederverwendung gemeinsamer Funktionen
- Konfiguration von Code trennen
- Abhängigkeiten verwalten
- Dokumentation und Bedienbarkeit
- wann ein Skript zu einer Anwendung wird

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	3 Tage
Durchführung	Präsenz oder Live Online, vorzugsweise als Inhouse-Schulung
Schwerpunkt	Inhouse-Schulungen für Unternehmen und Teams
Sprache	Deutsch, englische Fachbegriffe und Dokumentation
Unterlagen	Kursunterlagen, Beispiele und Übungsaufgaben
Technik	Eigener Rechner mit aktueller Python-Version und Entwicklungsumgebung

Inhouse-Schulungen

Die Schwerpunkte können an vorhandene Kenntnisse, eingesetzte Technologien und konkrete Anforderungen des Teams angepasst werden.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Der Kurs richtet sich an Entwicklerinnen und Entwickler, Administratoren, DevOps-Mitarbeiter und technisch orientierte Anwender, die wiederkehrende Aufgaben mit Python automatisieren möchten.

Er eignet sich insbesondere für Teilnehmer, die bereits kleinere Python-Skripte schreiben können und daraus zuverlässige Werkzeuge für den praktischen Einsatz entwickeln möchten.

Die Teilnehmer sollten grundlegende Python-Kenntnisse besitzen und mit folgenden Themen umgehen können:

- Variablen und grundlegende Datentypen
- Listen und Dictionaries
- Bedingungen und Schleifen
- Funktionen
- grundlegende Dateioperationen

Entwicklungserfahrung

Weitergehende Kenntnisse in Softwareentwicklung sind nicht erforderlich.

05 LERNZIELE

Was Sie nach dem Kurs können

Nach dem Kurs können die Teilnehmer:

- geeignete Aufgaben für eine Automatisierung identifizieren und strukturieren
- Dateien und Verzeichnisstrukturen automatisiert verarbeiten
- Text-, CSV- und JSON-Daten einlesen und transformieren
- externe Programme kontrolliert aus Python aufrufen
- eigene Kommandozeilenwerkzeuge entwickeln
- HTTP-Schnittstellen für Automatisierungsaufgaben nutzen
- Konfiguration und Zugangsdaten sinnvoll verwalten
- Fehlerfälle und Wiederanläufe berücksichtigen
- automatisierte Abläufe mit Logging nachvollziehbar machen
- wiederkehrende Skripte unbeaufsichtigt ausführen
- Automatisierungscode testbar und wartbar strukturieren
- aus einfachen Skripten robuste Werkzeuge für den praktischen Einsatz entwickeln

06 FAQ

Häufige Fragen

Ist der Kurs auch für Administratoren geeignet?

Ja. Der Schwerpunkt liegt auf praktischer Automatisierung und nicht auf Anwendungsentwicklung. Grundlegende Python-Kenntnisse sollten jedoch vorhanden sein.

Werden Linux und Windows behandelt?

Die verwendeten Python-Techniken sind weitgehend plattformunabhängig. Unterschiede zwischen Linux und Windows werden dort behandelt, wo sie für Dateisysteme, Prozesse oder zeitgesteuerte Aufgaben relevant sind.

Geht es auch um Web-Scraping oder Browser-Automatisierung?

HTTP-basierte Automatisierung wird behandelt. Browser-Automatisierung mit Werkzeugen wie Playwright oder Selenium ist dagegen ein eigenes Themengebiet und nicht Schwerpunkt dieses Kurses.

Werden REST APIs behandelt?

APIs werden aus Sicht der Automatisierung genutzt. Die Entwicklung eigener REST APIs ist Gegenstand von PY-006 „REST APIs mit Python und FastAPI“.

Können eigene Automatisierungsaufgaben eingebracht werden?

Ja. Bei Inhouse-Schulungen können typische wiederkehrende Aufgaben aus dem Arbeitsalltag des Teams als Beispiele oder Übungen aufgegriffen werden.

Ist der Kurs mit PY-001 kombinierbar?

Ja. Für Teilnehmer ohne ausreichende Python-Kenntnisse können Grundlagen und Automatisierung zu einer auf die Zielgruppe zugeschnittenen Schulung kombiniert werden.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de

<https://www.uc-it.de/coaching/python-automatisierung/>