

Testautomatisierung mit Python

Flexible und wartbare Testlösungen mit Python entwickeln

01 INTRO

Python eignet sich durch seine klare Syntax und sein umfangreiches Bibliotheksökosystem besonders gut für Testautomatisierung. Von API- und Integrationstests über Browserautomatisierung bis zu technischen Hilfswerkzeugen lassen sich damit kompakte und gleichzeitig professionelle Testlösungen entwickeln.

Der Kurs verbindet Python mit den Anforderungen moderner Testautomatisierung. Die Teilnehmer entwickeln automatisierte Tests mit pytest, strukturieren Fixtures und Testdaten und integrieren Web-, API- und technische Tests zu einer wartbaren Lösung.

DAUER

3 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Python-Grundkenntnisse und Grundlagen des Softwaretests

02 TRAININGSPROGRAMM

Inhalte

Python für Testautomatisierung

- Python-Ökosystem
- virtuelle Umgebungen
- Dependencies
- Projektstruktur
- Testbibliotheken
- Konfiguration
- typische Einsatzbereiche

Testen mit pytest

- Testfunktionen
- Assertions
- Test Discovery
- Marker
- Parametrisierung
- Exceptions
- Testselektion
- Kommandozeile

Fixtures

- Setup und Teardown
- Fixture Scope
- Abhängigkeiten
- Parametrisierung
- conftest.py
- Wiederverwendung
- saubere Testzustände

Testcode strukturieren

- Module
- Packages
- Helper
- Services
- Abstraktionen
- fachliche Assertions
- Wiederverwendung
- Wartbarkeit

Testdaten

- Parameter
- Dateien
- JSON
- Datenobjekte
- Builder und Factories
- Datenbanken
- dynamische Daten
- Cleanup

API-Testautomatisierung

- HTTP Clients
- Requests
- Responses
- JSON
- Assertions
- Authentifizierung
- Sessions
- wiederverwendbare API-Schichten

Web-Testautomatisierung

- Playwright
- Browser
- Locators
- Assertions
- Synchronisation
- Page Objects
- Fixtures
- Diagnose

Datenbank- und Integrationstests

- Datenbankzugriff
- Testdaten
- Transaktionen
- externe Services
- Mocks
- Stubs
- technische Integration
- Isolation

Mocking und Test Doubles

- unittest.mock
- Patch
- Mock
- Stub
- Rückgabewerte
- Fehler simulieren
- sinnvolle Grenzen
- Overmocking vermeiden

Konfiguration und Umgebungen

- Environment Variables
- Konfigurationsdateien
- Secrets
- unterschiedliche Zielsysteme
- lokale und CI-Ausführung
- reproduzierbare Umgebungen

CI/CD und Reporting

- pytest in Pipelines
- Marker
- parallele Ausführung
- Reports
- Logs
- Screenshots
- Artefakte
- Quality Gates

Praktisches Python-Automatisierungsprojekt

- Projekt aufsetzen
- pytest-Struktur entwickeln
- Fixtures und Daten integrieren
- API- oder Webtests automatisieren
- Tests stabilisieren
- Reports erzeugen
- CI-Ausführung vorbereiten

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 Python, Entwicklungsumgebung und pytest

Inhouse-Schulungen

Vorhandene Python-Projekte, APIs und Webanwendungen können als Testobjekt dienen.

04 ZIELGRUPPE

Für wen ist der Kurs gedacht?

Python-Entwickler, Tester, Quality Engineers und Testautomatisierer, die professionelle Testautomatisierung mit Python entwickeln möchten.

Voraussetzungen

Python-Grundkenntnisse und Grundlagen des Softwaretests.

05 LERNZIELE

Was Sie nach dem Kurs können

Die Teilnehmer können mit pytest strukturierte Testlösungen entwickeln, Fixtures und Testdaten beherrschen, Web- und API-Automatisierung integrieren und Tests in CI/CD ausführen.

06 FAQ

Häufige Fragen

Was unterscheidet den Kurs von Python Testing mit pytest?

Der Python-Kurs konzentriert sich auf **Softwaretests mit pytest für Python-Software**. Hier wird Python als **Werkzeug zum Aufbau einer Testautomatisierungslösung** betrachtet, die auch externe Webanwendungen, APIs, Datenbanken und Systeme testen kann.

Wird Playwright behandelt?

Ja, als Bestandteil Python-basierter Testautomatisierung. Der Playwright-Kurs behandelt die Technologie selbst wesentlich tiefer.

Muss ich pytest bereits kennen?

Nein. Python-Grundkenntnisse werden vorausgesetzt; pytest wird im Kurs aufgebaut.

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

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