

GUI-Entwicklung mit Python und Qt/PySide

Desktop-Anwendungen mit Python und Qt entwickeln

01 INTRO

Nicht jede Anwendung braucht ein Web-Frontend. Für technische Werkzeuge, interne Anwendungen und klassische Desktop-Software bietet Qt eine umfangreiche und plattformübergreifende GUI-Bibliothek. Mit PySide lässt sie sich direkt aus Python nutzen.

Der Kurs zeigt den Weg von einfachen Fenstern und Dialogen bis zu strukturierten Desktop-Anwendungen. Neben Widgets, Layouts und Events stehen die Trennung von Oberfläche und Programmlogik, Datenmodelle, Hintergrundverarbeitung und die Entwicklung wartbarer GUI-Anwendungen im Mittelpunkt.

DAUER

4 Tage

FORMAT

Präsenz · Live Online · Inhouse

VORKENNTNISSE

Sichere Python-Grundkenntnisse

02 TRAININGSPROGRAMM

Inhalte

Einstieg in Qt und PySide

- Qt und PySide
- Aufbau einer Qt-Anwendung
- QApplication
- Widgets und Fenster
- Event Loop
- grundlegender Lebenszyklus einer GUI-Anwendung
- PySide-Projekt einrichten und starten

Fenster und Widgets

- QWidget
- QMainWindow
- Labels und Eingabefelder
- Buttons
- Auswahlfelder
- Listen und Tabellen
- Dialoge
- Standarddialoge von Qt
- Eigenschaften und Zustände von Widgets

Layouts

- Oberflächen ohne feste Positionen gestalten
- horizontale und vertikale Layouts
- Grid Layout
- verschachtelte Layouts
- Größenverhalten von Widgets
- Abstände und Ausrichtung
- anpassbare Fenstergrößen

Signals und Slots

- ereignisgesteuerte Programmierung
- Signals und Slots
- eingebaute Signals
- Slots implementieren
- Parameter übertragen
- eigene Signals definieren
- Komponenten lose miteinander verbinden

Menüs, Toolbars und Actions

- Menüs
- Kontextmenüs
- Toolbars
- QAction
- Tastaturkürzel
- Statusleiste
- gemeinsame Actions für unterschiedliche Bedienelemente

Dialoge und Benutzerinteraktion

- modale und nichtmodale Dialoge
- eigene Dialoge
- Eingaben validieren
- Meldungen und Rückfragen
- Dateiauswahl
- Fortschrittsanzeigen
- Benutzeraktionen kontrolliert abbrechen

Qt Designer

- Oberflächen visuell erstellen
- Widgets und Layouts im Designer
- .ui-Dateien
- Designer-Oberflächen aus Python verwenden
- generierter Code oder dynamisches Laden
- Oberfläche und Programmlogik getrennt halten

Model/View-Architektur

- Daten und Darstellung trennen
- Model/View-Konzept von Qt
- List-, Table- und Tree-Views
- Standardmodelle
- eigene Modelle
- Daten aktualisieren
- Auswahl und Bearbeitung
- größere Datenmengen darstellen

Anwendungsarchitektur

- GUI und Fachlogik trennen
- Verantwortlichkeiten von Fenstern und Dialogen
- Controller- und Service-Komponenten
- Zustände verwalten
- Abhängigkeiten reduzieren
- wiederverwendbare GUI-Komponenten
- wachsende Anwendungen strukturieren

Dateien und Einstellungen

- Dateien öffnen und speichern
- Benutzereinstellungen
- QSettings
- zuletzt verwendete Dateien
- Anwendungszustand speichern
- Fensterpositionen und -größen wiederherstellen
- Konfiguration von Programmdateien trennen

Länger laufende Aufgaben

- warum die Oberfläche blockiert
- Event Loop und blockierender Code
- Threads in GUI-Anwendungen
- QThread
- Worker Objects
- Signals zwischen Threads
- Fortschritt melden
- Aufgaben abbrechen
- Thread-Sicherheit berücksichtigen

Fehlerbehandlung und Logging

- Exceptions in GUI-Anwendungen
- Fehler benutzerfreundlich anzeigen
- technische Details protokollieren
- Logging integrieren
- unerwartete Fehler behandeln
- Diagnoseinformationen bereitstellen

GUI-Anwendungen testen

- Programmlogik unabhängig von der Oberfläche testen
- testbare GUI-Architektur
- Widgets automatisiert testen
- Benutzerinteraktionen simulieren
- Signals überprüfen
- pytest und Qt im Überblick
- Grenzen von GUI-Tests

Anwendung bereitstellen

- Abhängigkeiten einer Desktop-Anwendung
- virtuelle Umgebungen
- Ressourcen und Konfigurationsdateien
- Icons und andere Assets
- Anwendungen paketieren
- PyInstaller im Überblick
- Besonderheiten von Windows, Linux und macOS

03 DURCHFÜHRUNG

Rahmen und Organisation

Dauer	4 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 Python-Entwicklerinnen und -Entwickler, die grafische Desktop-Anwendungen, technische Werkzeuge oder interne Benutzeroberflächen mit Qt entwickeln möchten.

Er eignet sich sowohl für Teilnehmer ohne bisherige GUI-Erfahrung als auch für Entwickler, die bereits kleinere PySide- oder PyQt-Anwendungen erstellt haben und diese systematischer strukturieren möchten.

Die Teilnehmer sollten mit folgenden Themen sicher umgehen können:

- grundlegende Python-Syntax
- Funktionen
- Module und Packages
- Klassen und Objekte
- Exceptions

GUI-Erfahrung

Erfahrungen mit Qt, PySide oder anderen GUI-Frameworks sind nicht erforderlich.

05 LERNZIELE

Was Sie nach dem Kurs können

Nach dem Kurs können die Teilnehmer:

- Desktop-Anwendungen mit PySide und Qt erstellen
- Fenster, Widgets und Dialoge sinnvoll einsetzen
- Benutzeroberflächen mit Layouts flexibel gestalten
- ereignisgesteuerte Abläufe mit Signals und Slots implementieren
- Menüs, Toolbars und Actions aufbauen
- Oberflächen mit Qt Designer erstellen und integrieren
- Daten mit der Model/View-Architektur darstellen
- GUI und Fachlogik voneinander trennen
- Einstellungen und Anwendungszustände verwalten
- länger laufende Aufgaben ausführen, ohne die Oberfläche zu blockieren
- Fehlerbehandlung und Logging in Desktop-Anwendungen integrieren
- GUI-Anwendungen testbar strukturieren
- Python-Desktop-Anwendungen für die Weitergabe vorbereiten

06 FAQ

Häufige Fragen

Wird PySide oder PyQt verwendet?

Der Kurs verwendet PySide. Die grundlegenden Qt-Konzepte sind weitgehend identisch, sodass sich viele Inhalte auch auf PyQt übertragen lassen.

Muss ich Qt bereits kennen?

Nein. Der Kurs beginnt mit dem Aufbau einer Qt-Anwendung und führt schrittweise zu komplexeren Oberflächen und Anwendungsstrukturen.

Wird Qt Designer verwendet?

Ja. Neben der Erstellung von Oberflächen direkt im Python-Code wird auch Qt Designer eingesetzt. Dabei wird insbesondere die saubere Trennung zwischen generierter Oberfläche und eigener Programmlogik behandelt.

Werden Threads behandelt?

Ja. Länger laufende Aufgaben sind bei Desktop-Anwendungen ein häufiges Problem. Deshalb werden QThread, Worker Objects und die Kommunikation über Signals praktisch behandelt.

Wird auch das Paketieren einer Anwendung behandelt?

Ja, im Überblick. Es wird gezeigt, welche Bestandteile für eine auslieferbare Desktop-Anwendung berücksichtigt werden müssen und wie beispielsweise PyInstaller eingesetzt werden kann.

Werden Datenbanken behandelt?

Eine Datenbank kann als Datenquelle in Beispielen verwendet werden, ist aber nicht Schwerpunkt des Kurses. Der systematische Datenbankzugriff mit Python ist Gegenstand von PY-007 „Python und Datenbanken“.

Können vorhandene Desktop-Anwendungen oder Anforderungen einbezogen werden?

Ja. Bei Inhouse-Schulungen können Beispiele und Schwerpunkte an vorhandene Anwendungen, technische Werkzeuge und Anforderungen des Teams angepasst werden.

UC IT Service · Ulrich Cuber · kontakt@uc-it.de
<https://www.uc-it.de/coaching/python-qt-pyside/>