Training Course on Advanced QGIS: Custom Tools and Plugins Development

Professional Development Training Course Outline

Category	GIS	**Duration**	10 Days
Base Fee	$3,000 USD	**Delivery Mode**	Online / On-site Hybrid

Course Introduction

Training Course on Advanced QGIS: Custom Tools and Plugins Development is meticulously designed for GIS professionals and developers seeking to unlock the full potential of QGIS through **customization** and **automation**. Participants will delve into the powerful **PyQGIS API**, mastering the art of developing **bespoke tools**, **plugins**, and **scripts** to streamline complex geospatial workflows. This program empowers users to transcend default functionalities, addressing specific organizational needs with tailored **geospatial solutions** and enhancing overall productivity in spatial data management and analysis.

Building upon foundational QGIS knowledge, this course emphasizes practical, hands-on application of **Python scripting** for **geoprocessing automation**, **user interface design** with PyQt, and integration of external libraries. Attendees will gain expertise in creating robust, scalable, and **reproducible GIS processes**, ultimately transforming QGIS into an even more potent platform for advanced spatial analysis, data visualization, and decision-making. Prepare to revolutionize your approach to **geospatial development** and become a QGIS power user.

Programme Curriculum

Training Course on Advanced QGIS: Custom Tools and Plugins Development

Introduction

Training Course on Advanced QGIS: Custom Tools and Plugins Development is meticulously designed for GIS professionals and developers seeking to unlock the full potential of QGIS through **customization** and **automation**. Participants will delve into the powerful **PyQGIS API**, mastering the art of developing **bespoke tools**, **plugins**, and **scripts** to streamline complex geospatial workflows. This program empowers users to transcend default functionalities, addressing specific organizational needs with tailored **geospatial solutions** and enhancing overall productivity in spatial data management and analysis.

Building upon foundational QGIS knowledge, this course emphasizes practical, hands-on application of **Python scripting** for **geoprocessing automation**, **user interface design** with PyQt, and integration of external libraries. Attendees will gain expertise in creating robust, scalable, and **reproducible GIS processes**, ultimately transforming QGIS into an even more potent platform for advanced spatial analysis, data visualization, and decision-making. Prepare to revolutionize your approach to **geospatial development** and become a QGIS power user.

Course Duration

10 days

Course Objectives

1. Gain a comprehensive understanding of the PyQGIS API for interacting with QGIS functionalities programmatically.
2. Design and implement bespoke geoprocessing tools to address specific analytical requirements.
3. Create, package, and distribute fully functional QGIS plugins for extended capabilities.
4. Develop Python scripts to automate repetitive GIS tasks and enhance efficiency.
5. Utilize PyQt and Qt Designer to craft intuitive graphical user interfaces for custom tools and plugins.
6. Seamlessly integrate third-party Python libraries for advanced data manipulation and analysis.
7. Interact with spatial databases (e.g., PostGIS) programmatically within QGIS.
8. Apply scripting for complex spatial analysis, including raster and vector operations.
9. Implement robust error handling and debugging techniques for stable plugin development.
10. Understand best practices for version control in collaborative plugin development.
11. Write efficient PyQGIS code for handling large datasets and complex operations.
12. Learn to effectively share and deploy custom tools and plugins within an organization
13. Understand the principles of contributing to the QGIS open-source ecosystem.

Organizational Benefits

- **Increased Efficiency & Automation:** Automate repetitive GIS tasks, significantly reducing manual effort and processing time.
- **Tailored Geospatial Solutions:** Develop custom tools that precisely meet unique organizational needs and workflows, enhancing productivity.
- **Cost Reduction:** Leverage the open-source nature of QGIS by developing in-house solutions, minimizing reliance on costly proprietary software licenses.
- **Enhanced Data Quality & Consistency:** Implement custom validation and processing routines to improve the accuracy and reliability of spatial data.
- **Improved Decision Making:** Facilitate faster and more informed decisions through streamlined data analysis and reporting.
- **Innovation & Competitive Advantage:** Foster a culture of innovation by enabling the creation of novel geospatial applications and methodologies.
- **Empowered Workforce:** Upskill GIS professionals, transforming them into advanced users and developers capable of tackling complex spatial challenges.
- **Reproducible Workflows:** Ensure consistency and reproducibility of analysis results through standardized custom tools and scripts.

Target Audience

1. Experienced QGIS Users and GIS Analysts

2. Geospatial Developers and Programmers
3. Data Scientists working with Spatial Data
4. Researchers and Academics utilizing GIS
5. Urban Planners and Environmental Scientists
6. Cartographers and Remote Sensing Specialists
7. Anyone seeking to automate and customize their QGIS workflows
8. IT Professionals supporting GIS infrastructure

Course Modules

Module 1: Introduction to PyQGIS and Development Environment Setup

- Understanding the QGIS architecture and PyQGIS API.
- Setting up your Python development environment (IDE, virtual environments).
- Navigating the QGIS Python Console for interactive scripting.
- Basic PyQGIS scripting for layer manipulation and data access.
- Introduction to QGIS processing framework for script integration.
- *Case Study:* Automating loading and styling of multiple shapefiles from a directory.

Module 2: Core PyQGIS Concepts: Layers, Features, and Geometries

- Accessing and managing vector layers (QgsVectorLayer).
- Working with features (QgsFeature) and attributes (QgsAttributes).
- Manipulating geometries (QgsGeometry) and spatial operations.
- Iterating through layers and features for batch processing.
- Handling Coordinate Reference Systems (CRS) in PyQGIS.
- *Case Study:* Developing a script to extract specific features based on attribute queries and save them to a new layer.

Module 3: Raster Data Processing with PyQGIS

- Understanding QGIS raster data model (QgsRasterLayer).
- Performing raster calculations and band manipulation.
- Extracting raster values at point locations.
- Georeferencing and resampling raster datasets programmatically.
- Integrating GDAL/OGR functionalities within PyQGIS.
- *Case Study:* Creating a script to perform a change detection analysis on two satellite imagery raster layers.

Module 4: Introduction to Plugin Development Fundamentals

- Understanding the QGIS plugin directory structure.
- Using Plugin Builder for initial plugin creation.
- Core components of a QGIS plugin: __init__.py, plugin.py, resources.qrc.
- Registering and running a basic "Hello World" plugin.
- Debugging plugins using print statements and external debuggers.
- *Case Study:* Building a simple plugin that adds a custom menu entry and displays a message box.

Module 5: Designing User Interfaces with PyQt and Qt Designer

- Introduction to PyQt and its role in QGIS UI development.
- Using Qt Designer to visually create .ui files.
- Converting .ui files to Python code for integration.

- Connecting UI elements (buttons, text boxes) to PyQGIS functions.
- Layout management for responsive plugin interfaces.
- *Case Study:* Designing a dialog box for a plugin that takes user input for a buffer distance.

Module 6: Integrating Processing Algorithms into Custom Tools

- Accessing and executing QGIS Processing algorithms via PyQGIS.
- Parameterizing Processing algorithms for dynamic execution.
- Creating custom processing scripts and integrating them into the toolbox.
- Batch processing with Processing algorithms.
- Handling results and output layers from Processing algorithms.
- *Case Study:* Developing a custom Processing script to perform a series of geoprocessing operations (e.g., clip, dissolve, buffer) on multiple input layers.

Module 7: Advanced User Interaction and Feedback

- Displaying messages and progress bars (QgsMessageBar, QgsProgressDialog).
- Handling user selections and input from the map canvas.
- Creating custom toolbars and dock widgets.
- Implementing interactive map tools for data capture.
- Providing user feedback and error notifications.
- *Case Study:* Building an interactive plugin that allows users to select features on the map and view their attributes in a custom dock widget.

Module 8: Working with Spatial Databases (PostGIS Integration)

- Connecting to PostGIS databases from PyQGIS.
- Executing SQL queries against spatial tables.
- Loading and saving data to and from PostGIS.
- Managing spatial indexes and database connections.
- Automating data import/export processes with PostGIS.
- *Case Study:* Developing a plugin to synchronize local QGIS layers with a PostGIS database, including handling new additions and updates.

Module 9: Advanced Plugin Features: Settings and Internationalization

- Storing and retrieving plugin settings (QgsSettings).
- Implementing internationalization for multi-language support.
- Adding custom icons and resources to plugins.
- Creating and managing temporary layers within plugins.
- Best practices for plugin maintainability and extensibility.
- *Case Study:* Modifying an existing plugin to allow users to configure output directory and add language support.

Module 10: Geometry Editing and Digitizing Tools

- Programmatic access to QGIS editing tools.
- Creating and modifying vector geometries using PyQGIS.
- Implementing custom digitizing and snapping tools.
- Handling undo/redo functionality in editing sessions.
- Developing tools for topological editing and validation.
- *Case Study:* Building a plugin that helps users rapidly digitize linear features by snapping to existing nodes and automatically calculating length.

Module 11: Web Mapping Integration and API Consumption

- Consuming OGC Web Services (WMS, WFS) with PyQGIS.
- Integrating external APIs for real-time data access (e.g., weather, traffic).
- Displaying web-based content within QGIS plugins.
- Working with GeoJSON and other web-friendly data formats.
- Considerations for offline data synchronization.
- *Case Study:* Creating a plugin that fetches weather data from an API and displays it as a thematic layer on the QGIS canvas.

Module 12: Data Visualization and Cartography with PyQGIS

- Programmatic control over layer symbology and styling.
- Creating rule-based and categorized renderers.
- Automating map layout and print composer generation.
- Generating map series using the Atlas functionality.
- Exporting maps to various formats (PDF, image).
- *Case Study:* Developing a script to automatically generate and export a series of thematic maps for different administrative units, each with a consistent layout.

Module 13: Debugging, Testing, and Performance Optimization

- Advanced debugging techniques for PyQGIS code.
- Writing unit tests for plugin functionalities.
- Profiling PyQGIS scripts for performance bottlenecks.
- Strategies for optimizing large dataset processing.
- Error logging and reporting mechanisms.
- *Case Study:* Identifying and resolving performance issues in a slow-running geoprocessing script, then implementing unit tests to ensure future stability.

Module 14: Plugin Distribution and Deployment

- Packaging plugins for distribution (ZIP files).
- Understanding the QGIS plugin repository and direct installation.
- Strategies for managing plugin updates and versions.
- Creating installation scripts for custom environments.
- Security considerations in plugin development.
- *Case Study:* Preparing a developed plugin for distribution, including documentation and a clear installation guide.

Module 15: Advanced Topics & Future Trends in QGIS Development

- Integrating machine learning libraries with PyQGIS.
- Leveraging QGIS Server for web-based GIS applications.
- Exploring QGIS 3D capabilities programmatically.
- Contributing to the QGIS open-source project.
- Emerging trends in geospatial development and QGIS.
- *Case Study:* Discussion and conceptualization of a potential future QGIS plugin that integrates AI for automated feature extraction from imagery.

Training Methodology

This course employs a **blended learning approach** combining interactive lectures, live demonstrations, and extensive hands-on exercises. Participants will engage in **practical coding**

sessions with guided mentorship. Each module incorporates **real-world case studies** and problem-solving scenarios to ensure immediate applicability of learned skills. Group discussions, peer code reviews, and dedicated Q&A sessions will foster a collaborative learning environment. The emphasis is on **project-based learning**, where attendees will incrementally build and refine their own custom QGIS tools and plugins throughout the course.

Register as a group from 3 participants for a Discount

Send us an email: info@fineskilltrainingcenter.org or call *+254769199797*

Certification

Upon successful completion of this training, participants will be issued with a globally- recognized certificate.

Upcoming Scheduled Sessions

Start Date	End Date	Location	Price
23 Nov 2026	04 Dec 2026	Online	$2,000
23 Nov 2026	04 Dec 2026	Physical	$3,000
23 Nov 2026	04 Dec 2026	Physical	$6,000
23 Nov 2026	04 Dec 2026	Physical	$6,000
23 Nov 2026	04 Dec 2026	Physical	$12,000
23 Nov 2026	04 Dec 2026	Physical	$9,000
23 Nov 2026	04 Dec 2026	Physical	$12,000
23 Nov 2026	04 Dec 2026	Physical	$12,000

Ready to enroll or request a customized program? Book online or contact us:

Register Online Now

Email: info@fineskilltrainingcenter.com | Website: www.fineskilltrainingcenter.com