

Inhaltsverzeichnis

Vorwort	9
1 Einführung in die Programmierung mit PyQt und PySide	11
1.1 Vorbereitungen	11
1.1.1 Entwicklungsumgebung	12
1.1.2 Python und PyQt bzw. PySide installieren	13
1.2 Eine erste Anwendung	14
1.3 Das Hauptfenster	18
1.4 Das Anwendungsmenü	20
1.4.1 Einfache Menüs mit Untermenüs	20
1.4.2 Verschachtelte Menüs und erweiterte Menüeinträge	22
1.5 Layouts	24
1.5.1 Freie Layouts	24
1.5.2 Horizontales und vertikales Layout	26
1.5.3 Elemente im Raster	29
1.5.4 Verschachteln von Layouts	31
1.5.5 Stapellayouts	32
1.5.6 Tab-Widget	33
1.6 Signale und Slots	35
1.6.1 Signale und Slots für das Stapellayout	38
1.7 Dialoge	40
1.7.1 Dialoge für Nachrichten und Fragen	41
1.7.2 Datei- und Verzeichnisdialoge	44
1.7.3 Eingaben abfragen	47
1.7.4 Eigene Dialoge erstellen	50

1.8	Event-Handler	54
1.8.1	Spezielle Event-Handler	55
1.8.2	Der allgemeine Event-Handler	59
1.8.3	Event-Filter	59
1.9	Qt-Klassen und -Objekte	61
1.9.1	QObject, QWidget und abgeleitete Klassen	61
1.9.2	QString, QVariant, Container-Klassen und andere	62
1.9.3	Objekthierarchie und Speicherverwaltung	63
1.10	Übersicht über die Qt-Dokumentation	65
2	Die Qt-Tools und Python	67
2.1	Internationalisierung und Lokalisierung	68
2.1.1	Quellcode für die Übersetzung vorbereiten	70
2.1.2	Übersetzungsdateien erzeugen	71
2.1.3	Übersetzen mit dem Qt Linguist	73
2.1.4	Übersetzungen im Quellcode laden	76
2.1.5	Übersetzungen und Kontext	80
2.2	Qt Designer	82
2.2.1	GUI der Beispielanwendung mit dem Qt Designer entwerfen	82
2.2.2	GUI in das Hauptprogramm integrieren	89
2.2.3	Lokalisierung der Qt-Designer-GUI	93
2.2.4	Der Anwendung weitere Dialoge hinzufügen	94
2.3	Ressourcen	96
2.3.1	Erstellung der Ressourcendatei	97
2.3.2	Ressourcen in einer Python-Anwendung verwenden	100
3	Projektstruktur und Paketerstellung	103
3.1	Projektstruktur für PyQt- und PySide-Anwendungen	104
3.1.1	Ordner und Dateien des Projekts	105
3.1.2	Die Dateien der Ordner bin und src	107
3.2	Automatisierung von rcc und uic	110
3.3	Erstellung von Quelltext-Paketen	112
3.4	Erstellung von Linux-Paketen	114
3.4.1	RPM-Pakete	116

3.4.2	Debian-Pakete	117
3.4.3	Debian-Paketerstellung mit PySide-Assistent	121
3.5	Erstellung einer Windows-Version samt Installer	124
3.5.1	Kompilierung der Anwendung	125
3.5.2	Erstellung des Installationspakets	127
4	Qt Quick und QML	133
4.1	Überblick über Qt Quick	134
4.1.1	Deklaratives Programmieren mit QML	135
4.1.2	Der QML Viewer	138
4.2	QML-Anwendungen	139
4.2.1	Ein erster View in QML	140
4.2.2	Die Daten als Modell	141
4.2.3	Eigene GUI-Elemente komponieren	143
4.2.4	Signale und Zustände	146
4.2.5	Zustandsänderungen und View-Wechsel	148
4.3	Vordefinierte QML-Elemente	150
4.4	QML und Python	153
4.4.1	QML-Views in Python-Anwendungen einbetten	153
4.4.2	Python-Daten in QML-Views darstellen	159
4.4.3	Python-Code aus QML aufrufen und umgekehrt	166
5	WebKit	173
5.1	Ein Webbrowser in Python	174
5.1.1	Erstellen der Oberfläche mit dem Qt Designer	175
5.1.2	Der Browser in Python	178
5.2	Cookies verwalten per Netzwerkmanager	180
5.3	WebKit in QML	186
5.3.1	Hauptanwendung in Python	187
5.3.2	Der QML-View der Anwendung	189
6	Grafik- und Audioprogrammierung	199
6.1	Grafikprogrammierung mit Szenen und Views	200
6.1.1	Hauptanwendung und Hauptfensterklasse mit View und Szene	202

6.1.2	Ein Foto-Browser als grafische Szene	205
6.1.3	Der Foto-Betrachter als eigener Dialog	210
6.2	Audioausgabe mit QtMultimedia	211
6.2.1	Abfrage der Hardware und unterstützten Codecs . . .	212
6.2.2	Audioplayer zum Abspielen von WAV-Dateien	213