Python GUI with PyQt

Learn to build modern and stunning GUIs in Python with PyQt5 and Qt Designer

Saurabh Chandrakar

Dr. Nilesh Bhaskarrao Bahadure

www.bpbonline.com

First published: 2024

Published by BPB Online
WeWork
119 Marylebone Road
London NW1 5PU

UK | UAE | INDIA | SINGAPORE

ISBN 978-93-55515-575

www.bpbonline.com

Dedicated to

My Parents
Dr. Surendra Kumar Chandrakar *and*
Smt. Bhuneshwari Chandrakar
brother
Shri Pranav Chandrakar,
Sister in law
Smt. Silky Chandrakar
to my wife
Smt. Priyanka Chandrakar
and to my lovely son
Master Yathartha Chandrakar

- Saurabh Chandrakar

My Parents
Smt. Kamal B. Bahadure *and* ***Late Bhaskarrao M. Bahadure***
to my in-laws
Smt. Saroj R. Lokhande *and* ***Shri Ravikant A. Lokhande***
and to my wife
Shilpa N. Bahadure
and to beautiful daughters
Nishita *and* ***Mrunmayee***
and to all my beloved students

- Dr. Nilesh Bhaskarrao Bahadure

About the Authors

- **Saurabh Chandrakar** is a Research & Development Engineer (Dy. Manager) at **Bharat Heavy Electricals Limited** (BHEL) Hyderabad. He is the winner of the best executive award on Operations Division by BHEL Hyderabad. Recently, he has been awarded the prestigious BHEL Excellence Award under Anusandhan category for Redundant Composite Monitoring System of Power Transformers project. He has 23 copyrights and 3 patent granted. Additionally, he has 4 patents filed. Moreover, he has published 5 books in reputed publications such as BPB New Delhi (*Programming Techniques using Python, Python for everyone. Building Modern GUIs with tkinter and python*), Scitech Publications Chennai (*Programming Techniques using matlab*) and IK International publishers (*Microcontrollers and Embedded System Design*). Additionally, he has also launched one video course on BPB titled *"First Time Play with Basic, Advanced Python concepts and complete guide for different python certification exams all in one umbrella."*
- **Dr. Nilesh Bhaskarrao Bahadure** received the Bachelor of Engineering degree in Electronics Engineering in 2000, the Master of Engineering degree in Digital Electronics in 2005, and the Ph.D. degree in Electronics in 2017 from KIIT Deemed to be University, Bhubaneswar, India. He is currently an Associate Professor in the Department of Computer Science and Engineering at Symbiosis Institute of Technology (SIT), Nagpur, Symbiosis International (Deemed University) (SIU), Pune, Maharashtra, India. He has more than 20 years of experience. Dr. Bahadure is a life member of IE(I), IETE, ISTE, ISCA, SESI, ISRS, and IAENG professional organizations. He has published more than 40 articles in reputed international journals and conferences, and has five books to his credit. He is the reviewer of many indexed journals such as IEEE Access, IET, Springer, Elsevier, Wiley and so on. His research interests are in the areas of Sensor Technology, the Internet of Things, and Biomedical Image Processing.

About the Reviewer

Prasenjeet Damodar Patil is currently working as an Associate Professor at the School of Computing, M.I.T A.D.T University, Pune. He has completed his B.E. in E&TC Engg. and M. Tech. in Electronics from Walchand College of Engineering Sangli, India. He did his Ph.D. in E&TC Engineering in 2018. Prasanjeet has 14+ years of teaching experience and has published more than 16 papers in reputed Journals. His research interest includes Computational Electromagnetics applications in Integrated Optics, IoT, and digital Image Processing.

Acknowledgements

- First and foremost, I would like to thank you all for selecting this book. It has been written with the beginner reader in mind. First of all, I take this opportunity to greet and thank my mentor "Prof. Nilesh Bahadure Sir" for motivating me and always communicating his expertise fully on topics related to Python. I am very thankful for being his protégé. I appreciate his belief in me for always standing behind me and pushing me to achieve more. The phrase "Journey of a Thousand Miles Begins with a Single Step" is something he always reminds me of.

  My parents, Dr. Surendra Kumar Chandrakar and Smt. Bhuneshwari Chandrakar, my brother, Shri Pranav Chandrakar, my sister-in-law Silky Chandrakar, my beloved wife, Mrs. Priyanka Chandrakar, my adorable son Yathartha Chandrakar, and all of my friends have inspired me and given me confidence over the years. Last but not the least, I would like to express my sincere gratitude to the staff at BPB Publications for their contributions and insights that made parts of this book possible.

  — Saurabh Chandrakar

- It was my privilege to thank Dr. S. B. Mujumdar, Chancellor of the Symbiosis International University, Pune, and Shri. Vijay Kumar Gupta, Chairman of Beekay Industries Bhilai and BIT Trust, for his encouragement and support. I thank my mentors, Dr. Arun Kumar Ray, Dean, School of Electronics Engineering, KIIT Deemed to be University, Bhubaneswar, and Dr. Anupam Shukla, Director, SVNIT Surat. I would like to thank Dr. Vidya Yeravdekar, Principal Director of Symbiosis Society and the Pro Chancellor of Symbiosis International University, Pune, Dr. Ramakrishnan Raman, Vice Chancellor of the Symbiosis International University, Pune, and Dr.Ketan Kotecha, Dean, Faculty of Engineering, Symbiosis International University, Pune, and Dr. Nitin Rakesh, Director, SIT Nagpur, for their advice, and encouragement throughout the preparation of the book.

  I would also like to thank Dr. Sanjeev Khandal, HOD, Department of Aeronautical Engineering, SGU Kolhapur, my well-wisher Dr. Prasenjeet D. Patil, Associate Professor, MIT ADT University, Pune, and my colleagues in Symbiosis Institute of Technology Nagpur for providing valuable suggestions and lots of encouragement throughout the project.

I am thankful to Prof. Dr. N. Raju, Sr. Assistant Professor, SASTRA University, Thanjavur, Tamil Nadu, for his support, assistance during writing, and for his valuable suggestions.

I would also like to thank Dr. Ravi M. Potdar, Sr. Associate Professor, BIT Durg, and Dr. Md. Khwaja Mohiddin, Associate Professor, BIT Raipur, for providing valuable suggestions and encouragement throughout the project. Writing a beautiful, well-balanced, and acquainted book is not a work of one or two days or a month; it takes a lot of time and patience, as well as hours of hard work. Many thanks to my family members, my parents, wife, children, and well-wishers for their kind support. Without them and their faith and support, writing this classic book would have remained just a dream. I also like to thank my students, who have always been with me, for relating problems and finding solutions too. The perfection in any work does not come in a day. It needs a lot of effort, time, hard work, and sometimes, proper guidance.

It is my privilege to thank Prof. (Dr.) Ram Dhekekar, Professor, Department of Electronics & Telecommunication Engineering, SSGMCE Shegaon, and Dr. C. G. Dethe, Director UGC Staff College Nagpur. Last but not the least, I would like to offer an extra special thanks to the staff at “BPB Publications Private Limited" for their insight and contribution in polishing this book.

Most significantly, I want to thank Lord Ganesha for all of the work I was able to put into the book's preparation. I would not be as zealous as I am now if it weren't for God's amazing creation of the universe.

For since the creation of the world God’s invisible qualities - his eternal power and divine nature - have been clearly seen, being understood from what has been made, so that men are without excuse.”

— Dr. Nilesh Bhaskarrao Bahadure

Preface

The purpose of this book is to introduce readers with little to no programming experience in Python Graphical User Interface (GUI) to use Python binding for the GUI toolkit called PyQt5. A GUI application can be created in any programming language, whether VB.Net, C#.Net etc. In this book, we will see how to create a GUI application using PyQt5. The readers will get the foundational knowledge and skills necessary to start writing code for creating a GUI app in Python language. We will use a Graphical tool for creating user interfaces in Qt called Qt Designer. By mastering PyQt5, readers can apply this knowledge to solve real-world problems and create various useful applications according to their needs.

The first part of the book covers PyQt5 library and the overall layout of Qt Designer tool. Then we will see some insights into layout management and the concept of event-driven programming and its implementation in the context of Python programming with the usage of the concept of signals and slots. Finally, in the later part of the book, we shall see an insight into various widgets related to buttons, container item views, container, input, and display widgets.

This book covers a wide range of topics, from the basic definitions of different widgets to various solved examples with well-explanatory code. Overall, the book provides a solid foundation for beginners to start their journey for getting trained in Python GUI using the PyQt5 library along with the Qt Designer layout tool.

This book is divided into **9 chapters**. Each chapter description is listed as follows.

Chapter 1: Introduction to PyQt5 and Qt Designer Tool – This chapter will start by comparing powerful and cross-platform toolkit graphical PyQt5 with the tkinter library. You will learn how to install PyQt5 framework installation along with the creation of a basic GUI form using PyQt5 without and with using the class. We shall explore the components inside Qt Designer along with different pre-defined templates. In the latter half of the chapter, we will create a user credential app initially focusing on view in Qt Designer (.ui file, which is an XML file), then converting it into a python code (.py) using pyuic5 command and finally creating a new python file which will be importing the python code for user interface design and adding some useful logic to create a basic Login application for the user.

Chapter 2: Getting Insights of Layout Management – will cover the concept on widgets placement using absolute positioning approach. We shall see widgets placement using layout classes where initially we will see how to organize widgets either horizontally or vertically

using QBoxLayout class. We will explore how to arrange the widgets in a row side by side using QHBoxLayout with the usage of methods viz addStretch, addWidget, addLayout. Alternatively, we will look into arranging the widgets vertically with addStretch method using QVBoxLayout. Then we shall find out arranging widgets in a grid of rows and columns using QGridLayout. Also, we will look into creating an application using QFormLayout. In the end, we will be confident to create " User Credential App " application using absolute positioning , QBoxLayout, QGridLayout and QFormLayout class.

Chapter 3: Getting Insights of Events, Signals and Slots – will explore the concept of event-driven programming and how it is implemented in the context of Python programming using signals and slots. The above chapter will guide us through utilizing signals and slots to create and manage events. We will discover how to define signals and connect them to slots so that our program can respond to events in a useful way. We will also investigate various events and signals and how to use them to create interactive user interfaces or handle external inputs.

Chapter 4: Getting Insights of Button Widgets in Qt Designer – will cover the concept of button widgets which are commonly used for creating interactive user interfaces and make them understand their properties, functionality, and customization options. By exploring the features and settings related to button widgets, user can effectively design and implement user-friendly interfaces in our Qt applications. Qt Designer offers a number of different button widgets, including CheckBox, Push Button, Tool Button, Radio Button, Command Link Button etc and shall explore description, properties, important methods , important signals and an application example with output display of each button widget in detail. The important properties of QObject, QWidget and QAbstractButton will be covered with images at the end as an addon information.

Chapter 5: Getting Insights of Item Views in Qt Designer – will cover the concept of Item Views in Qt Designer which are commonly used for creating user interactive interfaces. We will look into their properties, functionality, and customization options. We will explore Qt Designer's item view widgets, like QTableView, QTreeView, and QListView, which are effective tools for presenting data in an organised and ordered way. User can use these widgets to display data in simple list formats (QListView), hierarchical tree structures (QTreeView), or rows and columns (QTableView).

Chapter 6: Getting Insights of Item Widgets (Item-Based) in Qt Designer – will cover the concept of Item Widgets in Qt Designer which are commonly used for creating interactive user interfaces. We shall look to understand their properties, functionality, customization options and also will gain a thorough grasp of how to create and operate item-based widgets within the Qt Designer environment. User will be able to create dynamic, interactive user interfaces using the power of item-based widgets and will also discover different features and characteristics to

personalize the look and behavior of widgets viz list widget, tree widget and table widget. In order to facilitate user interaction and implement functionality, user will learn to manage the events and signals connected to item-based widgets.

Chapter 7: Getting Insights of Containers in Qt Designer – will cover the concept of container widgets offered by Qt Designer, their features, and how to customize them to design aesthetically pleasing and user-friendly interfaces. User will investigate various container widget types and will obtain knowledge of the precise features and capabilities provided by each widget. They will get to know what container widgets are and how they work, its different types, its usage to create layouts and shall also explore customizing appearance of container widgets.

Chapter 8: Getting Insights of Input Widgets in Qt Designer – will deal with concept of many input widgets available and how to utilize them effectively to create interactive user interfaces. Users will have a solid understanding of input widgets including QLineEdit, QSpinBox, QComboBox, QTextEdit etc. as well as their corresponding characteristics, functionality, and customization choices. They all shall experience the knowledge needed to incorporate these input widgets into their designs so that they may enter data, select options, and interact with the program. The use of input validation methods, processing user input events, and linking signals and slots to accomplish required functionality are also covered for the benefit of the readers. In the end, they will be able to develop simple, user-friendly interfaces which will effectively collect user input and deliver a smooth user experience by mastering input widgets in Qt Designer.

Chapter 9: Getting Insights of Display Widgets in Qt Designer – will explain the concept of display widgets in Qt Designer. Initially we will learn how to display static text or images using labels and how to change their font, color, alignment, and size. And then understand how labels can improve the GUI's visual presentation of information. We will examine the TextBrowser widget's features to learn how to display and control rich text content. Also, we will learn how to add hyperlinks, graphics, and formatting choices to text displays to make them dynamic and interactive. Next, we will explore how to add a calendar widget to our GUI application. We shall discover how to customize the calendar widget's look, structure, and behavior to meet specific application requirements. We will explore how to display numerical values, such as counters using the LCDNumber widget and also how to modify the LCDNumber widget's digit count, decimal accuracy, look, and style. Finally, we will look into exploring ProgressBar widget to show how a task or operation is progressing. We will learn how to dynamically update the progress bar based on our application.

Code Bundle and Coloured Images

Please follow the link to download the
Code Bundle and the ***Coloured Images*** of the book:

https://rebrand.ly/s2nzbop

The code bundle for the book is also hosted on GitHub at:
https://github.com/bpbpublications/Python-GUI-with-PyQt.
In case there's an update to the code, it will be updated on the existing GitHub repository.

We have code bundles from our rich catalogue of books and videos available at **https://github.com/bpbpublications**. Check them out!

Errata

We take immense pride in our work at BPB Publications and follow best practices to ensure the accuracy of our content to provide with an indulging reading experience to our subscribers. Our readers are our mirrors, and we use their inputs to reflect and improve upon human errors, if any, that may have occurred during the publishing processes involved. To let us maintain the quality and help us reach out to any readers who might be having difficulties due to any unforeseen errors, please write to us at :

errata@bpbonline.com

Your support, suggestions and feedbacks are highly appreciated by the BPB Publications' Family.

Piracy

If you come across any illegal copies of our works in any form on the internet, we would be grateful if you would provide us with the location address or website name. Please contact us at **business@bpbonline.com** with a link to the material.

If you are interested in becoming an author

If there is a topic that you have expertise in, and you are interested in either writing or contributing to a book, please visit **www.bpbonline.com**. We have worked with thousands of developers and tech professionals, just like you, to help them share their insights with the global tech community. You can make a general application, apply for a specific hot topic that we are recruiting an author for, or submit your own idea.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions. We at BPB can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about BPB, please visit **www.bpbonline.com**.

Join our book's Discord space

Join the book's Discord Workspace for Latest updates, Offers, Tech happenings around the world, New Release and Sessions with the Authors:

https://discord.bpbonline.com

Table of Contents

1. Introduction to PyQt5 and Qt Designer Tool 1

Introduction 1
Structure 1
Objectives 2
Comparison of PyQt5 with tkinter library 2
PyQt5 framework installation 4
First GUI form creation using PyQt5 without using class 5
GUI form creation using PyQt5 by using class 8
Installation of Qt Designer with pre-defined templates 13
Components of Qt Designer 16
User credential app demo 17
Conclusion 29
Points to remember 29
Questions 30

2. Getting Insights of Layout Management 31

Introduction 31
Structure 32
Objectives 32
Widgets placement using absolute positioning 32
Widgets placement using layout classes 37
QBoxLayout 37
QHBoxLayout 37
QVBoxLayout 47
QGridLayout 54
Basic QGridLayout 54
QGridLayout span 57
QGridLayout Stretch 59
QFormLayout 63
Conclusion 77

Points to remember....78
Questions....78

3. Getting Insights of Events, Signals and Slots....79
Introduction....79
Structure....79
Objectives....80
Introduction to Events, Signals, and Slots....80
Usage of Toolbar icons in Qt Designer....92
Signal slot examples in Qt Designer....93
Conclusion....106
Points to remember....106
Questions....107

4. Getting Insights of Button Widgets in Qt Designer....109
Introduction....109
Structure....110
Objectives....110
Push Button....110
Important properties....111
autoDefault....111
default....111
flat....112
Important methods....112
Important signals....112
Tool Button....121
Important properties....121
popupMode....121
toolButtonStyle....122
autoRaise....122
arrowType....122
Important methods....123
Important signals....123

Radio Button 131
Important properties *131*
Important methods *132*
Important signals *132*
Check Box 136
Important properties *136*
tristate *136*
Important methods *137*
Important signals *137*
Command Link Button 144
Important properties *144*
Important methods *144*
setDescription *144*
Important signals *145*
Dialog Button Box 147
Important properties *148*
orientation *148*
standardButtons *148*
centerButtons *149*
Important methods *149*
Important signals *150*
Common properties for Button widgets 153
objectName *154*
enabled *154*
geometry *154*
sizePolicy *155*
minimumSize *156*
maximumSize *156*
sizeIncrement *157*
baseSize *157*
palette *157*
font *158*

cursor *158*
mouseTracking *158*
tabletTracking *159*
focusPolicy *159*
contextMenuPolicy *159*
acceptDrops *160*
toolTip *160*
toolTipDuration *160*
statusTip *161*
whatsThis *161*
accessibleName *161*
accessibleDescription *161*
layoutDirection *162*
autoFillBackground *162*
stylesheet *162*
locale *163*
inputMethodHints *163*
text *164*
icon *164*
iconSize *165*
shortcut *165*
checkable *165*
checked *166*
autoRepeat *166*
autoExclusive *166*
autoRepeatDelay *167*
autoRepeatInterval *167*
Conclusion 167
Points to remember 168
Questions 169

5. Getting Insights of Item Views in Qt Designer....171
Introduction....171
Structure....172
Objectives....172
Data presentation....172
List view....173
Important properties....173
movement....173
flow....174
isWrapping....174
resizeMode....174
layoutMode....174
spacing....175
gridSize....175
viewMode....175
modelColumn....175
uniformItemSizes....176
batchSize....176
wordWrap....176
selectionRectVisible....176
Important methods of QAbstractItemView base class....176
Important signal of QAbstractItemView base class....177
Tree view....179
Important properties....180
autoExpandDelay....180
indentation....180
rootIsDecorated....180
uniformRowHeights....180
itemsExpandable....181
sortingEnabled....181
animated....181
allColumnsShowFocus....181

wordWrap *182*
headerHidden *182*
expandsOnDoubleClick *182*
Table view 186
Important properties *187*
showGrid *187*
gridStyle *187*
sortingEnabled *187*
wordWrap *187*
cornerButtonEnabled *188*
Column view 191
Important properties *191*
resizeGripsVisible *191*
QFrame 194
frameShape *194*
frameShadow *195*
lineWidth *196*
midLineWidth *196*
QAbstractScrollArea 196
verticalScrollBarPolicy *196*
horizontalScrollBarPolicy *197*
sizeAdjustPolicy *197*
QAbstractItemView 197
autoScroll *198*
autoScrollMargin *198*
editTriggers *198*
tabKeyNavigation *199*
showDropIndicator *199*
dragEnabled *199*
dragDropOverwriteMode *200*
dragDropMode *200*
defaultDropAction *201*

alternatingRowColors *201*
selectionMode *201*
selectionBehavior *202*
iconSize *203*
textElideMode *203*
verticalScrollMode *203*
horizontalScrollMode *204*
QStandardItemModel 204
Conclusion 204
Points to remember 205
Questions 205

6. Getting Insights of Item Widgets (Item-Based) in Qt Designer 207
Introduction 207
Structure 208
Objectives 208
List widget 208
Important properties *208*
currentRow *208*
sortingEnabled *209*
Important methods *209*
Important signals *209*
Tree widget 218
Important properties *218*
Important methods *219*
Important signals *219*
Table widget 227
Important properties *227*
rowCount *228*
columnCount *228*
Important methods *228*
Important signals *229*
Conclusion 236

Points to remember....236
Questions....237

7. Getting Insights of Containers in Qt Designer....239
Introduction....239
Structure....240
Objectives....240
Group Box....240
Important properties....241
title....241
alignment....241
flat....241
checkable....241
checked....242
Important methods....242
Important signals....242
Scroll area....246
Important properties....246
widgetResizable....246
alignment....247
Important methods....247
Important signals....247
scrollContentsBy(int dx, int dy)....247
Toolbox....251
Important properties....251
currentIndex....251
currentItemText....251
currentItemName....251
currentItemIcon....252
currentItemToolTip....252
tabSpacing....252
Important methods....252
Important signals....253

currentChanged(index) *253*
Tab widget 256
Important properties *257*
tabPosition *257*
tabShape *257*
currentIndex *257*
iconSize *257*
elideMode *258*
userScrollButtons *258*
documentMode *258*
tabsClosable *258*
movable *259*
tabBarAutoHide *259*
currentTabText *259*
currentTabName *259*
currentTabIcon *259*
currentTabToolTip *260*
currentTabWhatsThis *260*
Important methods *260*
Important signals *261*
currentChanged *261*
tabCloseRequested *261*
Stacked widget 265
Important properties *265*
currentIndex *266*
Important methods *266*
Important signals *266*
currentChanged(arg_ _1) *266*
widgetRemoved *266*
Frame 269
Important properties *270*
Important methods *270*

Widget....272
Important properties....272
Important methods....273
Important signals....273
MDI Area....275
Important properties....275
background....275
activationOrder....275
viewMode....276
documentMode....276
tabsClosable....276
tabsMovable....276
tabShape....277
tabPosition....277
Important methods....277
Important signals....278
subWindowActivated(arg_ _1)....278
Dock widget....281
Important properties....281
floating....281
features....281
allowedAreas....282
windowTitle....282
dockWidgetArea....282
docked....283
Important methods....283
Important signals....283
Conclusion....286
Points to remember....286
Questions....287

8. Getting Insights of Input Widgets in Qt Designer 289
Introduction 289
Structure 290
Objectives 291
Combo Box 291
Important properties *291*
editable *291*
currentText *292*
currentIndex *292*
maxVisibleItems *292*
maxCount *292*
insertPolicy *292*
sizeAdjustPolicy *293*
minimumContentsLength *293*
iconSize *293*
duplicatesEnabled *293*
Frame *294*
modelColumn *294*
Important methods *294*
Important signals *294*
Font Combo Box 298
Important properties *298*
writingSystem *298*
fontFilters *298*
currentFont *298*
Important methods *299*
Important signals *299*
currentFontChanged(QFont) *299*
Line Edit 301
Important properties *302*
inputMask *302*
text *303*

maxLength *303*
frame *303*
echoMode *303*
cursorPosition *304*
alignment *304*
dragEnabled *304*
readOnly *304*
placeholderText *305*
cursorMoveStyle *305*
clearButtonEnabled *305*
Important methods *305*
Important signals *306*
TextEdit 313
Important properties *313*
autoFormatting *313*
tabChangeFocus *313*
documentTitle *313*
undoRedoEnabled *314*
lineWrapMode *314*
lineWrapColumnOrWidth *314*
readOnly *314*
html *314*
overwriteMode *315*
tabStopWidth *315*
tabStopDistance *315*
acceptRichText *315*
cursorWidth *316*
textInteractionFlags *316*
placeholderText *316*
Important methods *316*
Important signals *317*
Plain Text Edit 322

Important properties 322
plainText 322
maximumBlockCount 323
backgroundVisible 323
centerOnScroll 323
Important methods 324
Important signals 324
Spin Box 328
Important properties 328
wrapping 328
frame 329
alignment 329
readOnly 329
buttonSymbols 329
specialValueText 329
accelerated 330
correctionMode 330
keyboardTracking 330
showGroupSeparator 330
suffix 331
prefix 331
minimum 331
maximum 331
singlestep 332
value 332
displayIntegerBase 332
Important methods 332
Important signals 333
valueChanged(arg__1) 333
Double Spin Box 337
Important properties 337
decimals 337

Important methods *337*
setDecimals(prec) *338*
Important signals *338*
valueChanged(arg__1) *338*
Date/Time Edit 342
Important properties *342*
dateTime *342*
date *343*
time *343*
maximumDateTime *343*
minimumDateTime *343*
maximumDate *344*
minimumDate *344*
maximumTime *344*
minimumTime *344*
currentSection *345*
displayFormat *345*
calendarPopup *345*
currentSectionIndex *345*
timeSpec *346*
Important methods/signals *346*
Dial 350
Important properties *350*
minimum *350*
maximum *351*
singleStep *351*
pageStep *351*
value *351*
sliderPosition *351*
tracking *352*
orientation *352*
invertedAppearance *352*

invertedControls *352*
Properties of QDial *353*
wrapping *353*
notchTarget *353*
notchesVisible *353*
Important methods/signals *353*
QScrollBar 356
QSlider 361
Important properties *361*
tickPosition *361*
tickInterval *361*
Important methods/signals *361*
Important properties *362*
Key Sequence Edit 365
Important properties *365*
keySequence *365*
Important methods/signals *366*
Conclusion 367
Points to remember 368
Questions 368

9. Getting Insights of Display Widgets in Qt Designer 371
Introduction 371
Structure 371
Objectives 372
Introduction to the display widgets in Qt Designer 372
Label 372
Important properties *373*
text *373*
textFormat *373*
pixmap *373*
scaledContents *373*
alignment *374*

wordWrap *374*
margin *374*
indent *374*
openExternalLinks *374*
textInteractionFlags *375*
buddy *375*
Important methods/signals *375*
Text browser 378
Important properties *379*
Source *379*
searchPaths *379*
openExternalLinks *379*
openLinks *379*
Important methods/signals *380*
Calendar widget 382
Important properties *383*
selectedDate *383*
minimumDate *383*
maximumDate *384*
firstDayOfWeek *384*
gridVisible *384*
selectionMode *385*
horizontalHeaderFormat *385*
verticalHeaderFormat *385*
navigationBarVisible *385*
dateEditEnabled *386*
dateEditAcceptDelay *386*
Important methods/signals *386*
selectionChanged() *386*
LCD number 389
Important properties *389*
smallDecimalPoint *389*

digitCount *389*
mode *389*
segmentStyle *390*
value *390*
intValue *390*
Important methods/signals *391*
display(num) *391*
Progress bar 393
Important properties *394*
minimum *394*
maximum *394*
value *394*
alignment *394*
textVisible *395*
orientation *395*
invertedAppearance *395*
textDirection *395*
format *396*
Important methods/signals *396*
valueChanged(value) *396*
Conclusion 398
Points to remember 399
Questions 399

Index 401-409

Chapter 1
Introduction to PyQt5 and Qt Designer Tool

Introduction

In our previously launched book *Building Modern GUIs with tkinter and Python*, we learned how to create GUI forms using the Tk interface library. Now, we will be looking at another approach to create the same GUI forms using one of the most popular cross-platform libraries, known as PyQt5 library, which was developed by Riverbank Computing. A robust and cross-platform graphical toolkit called Qt has a Python binding called PyQt5. Python is a well-known and simple-to-learn programming language, and with PyQt5 developers, we can simply construct GUI applications using Python. With PyQt5, a visual layout tool called Qt Designer is included. Without having to write any code explicitly, it enables developers to construct GUI layouts quickly and easily by dragging and dropping widgets onto a canvas. Additionally, Qt Designer offers a selection of editable widgets that may be used to design distinctive user interfaces with eye-catching visuals.

Structure

In this chapter, we will discuss the following topics:

- Comparison of PyQt5 with tkinter library
- PyQt5 framework installation

- First GUI form creation using PyQt5 without using class
- GUI form creation using PyQt5 by using class
- Installation of Qt Designer tool with pre-defined templates
- Components of Qt Designer
- User Credential App Demo

Objectives

By the end of this chapter, the reader will be able to compare the powerful and cross-platform toolkit graphical PyQt5, with the tkinter library. We shall see how to install the PyQt5 framework installation, along with the creation of a basic GUI form using PyQt5, first without, and then using the class. Moreover, we will be looking into the components inside Qt Designer along with different pre-defined templates. Finally, we will be creating a user credential app, initially focusing on the view in Qt Designer (`.ui` file which is a XML file), then converting it into a Python code (`.py`) using `pyuic5` command, and finally creating a new Python file, which will be importing the Python code for user interface design and adding some useful logic to create a basic login application for the user.

Comparison of PyQt5 with tkinter library

There are numerous Qt classes in a set of Python modules, which are compatible with many operating systems such as iOS, Windows, Linux, Unix, Android, and so on, which is a part of the top-level Python package, that is, PyQt5 library. Qt5 in PyQt5 stands for **Qt version 5.** This library provides us the advantage of Python binding with the Qt C++ toolkit. An important point to note is that under the **GNU General Public License** (**GNU GPL** or **GPL**) v3 license, the PyQt5 license is being released. One may wonder that if we have already learned about tkinter, then why we should learn about PyQt5 as well? To better understand this, let us check the advantages of PyQt5 over other library packages such as tkinter:

- **Flexibility in coding**: In order to establish simple communication between objects, there is a concept of signals and slots, which gives us flexibility for GUI programming using PyQt5, when dealing with events.
- **More than GUI toolkit:** Using PyQt5, we can build entire applications using its graphics, printer support, networking, database access, and so on. It is like an application framework.
- **Numerous UI components:** PyQt5 offers numerous widgets such as `QLabel`, `QButton`, `QCombobox`, and so on, such that each widget has some basic image well suited in all the platforms. Numerous advanced widgets are also available on the topic in this library.

- **Numerous study resources:** If there is no documentation, then you might wonder what the point of learning is. PyQt5 comes with a rich array of documentation since it is one of the most commonly used Python packages for GUI creation.
- **Easy to understand:** We can easily use previous knowledge of either Python, Qt or C++, thus making PyQt5 easy to understand.
- **Preferred choice of GUI developers**: Due to simplicity and ease of use, many GUI developers opt for functionalities that come with PyQt5 to develop their own applications.
- **GUI widget appearance:** The appearance of PyQt5 is nice and pleasing to the eyes.

Now, we may wonder which library to choose to create GUI forms using PyQt5 or tkinter. This generally depends on user application and the willingness to learn and explore. *Table 1.1* further shows the differences between the libraries:

Fundamental points	PyQt5	tkinter
Licensing	Commercial license is applicable if not complying to provide an application to the end user under GPL license.	It is free if needed to supply commercially.
Library	Vast	Small compared to PyQt5.
Understanding time	More	Less
Widgets appearance	Nice and modern look	Orthodox and quite old-fashioned
Provision of advanced widgets	Yes	No
Interfacing with other designer tool	Yes, Qt Designer	Qt Designer is not there for tkinter
Default supply of inbuilt libraries	No, we need to install it separately	Yes, as it comes along with default standard Python library
Concept of signals and slots for communication	Yes	No

Table 1.1: *Differences between libraries PyQt5 and tkinter*

You can decide which library to choose to create GUI forms, depending on your application. We will later discuss how to create a **UI form** using Qt Designer in detail, as well as the widgets associated with it and the Python code with logic written by importing the auto-code generated by creating the UI form in detail.

PyQt5 framework installation

The Python version we will be using for discussing PyQt5 will be 3.7.3. (You can try working with new version as on today, that is, 3.11.5 dated Aug 24, 2023) From the location where we have installed our Python, first type the command **`python --version`** to check the version installed and then type the following command to install pyqt5:

```
pip install pyqt5
```

Refer to *Figure 1.1:*

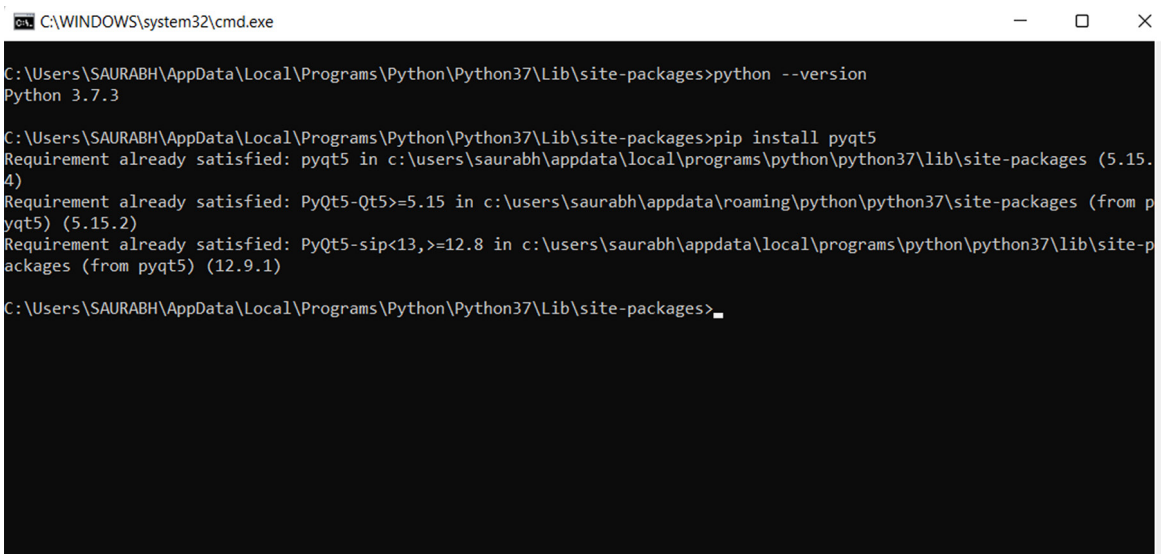

Figure 1.1: *PyQt5 installation*

Once installed, we can verify the installation of PyQt5 under Python site-packages folder, as shown in *Figure 1.2*:

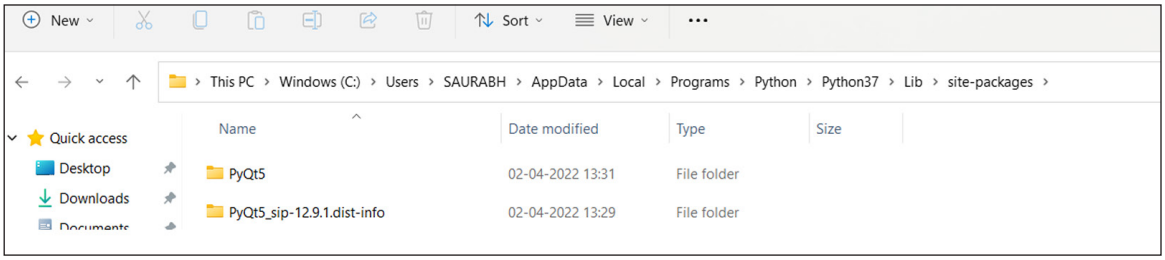

Figure 1.2: *PyQt5 folder*

We can check some of the modules present in PyQt5. If we do not get any error, then we can again cross verify that PyQt5 is successfully installed, as shown in *Figure 1.3:*

```
C:\Users\SAURABH>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from PyQt5 import QtGui
>>>
```

Figure 1.3: *PyQt5 package verification*

First GUI form creation using PyQt5 without using class

Now, we shall see a basic UI form creation using PyQt5 without using class. This is quite important. As you might have realized, most of the UI forms which we will be creating will be using the concept of class and object. It is highly recommended that you go through the concept of class before moving further in this chapter. It is well explained in our previous books, *Programming Techniques using Python* and *Python for Everyone*. Here, we will only focus on our basic UI form creation. So, without any further delay, let us start. Refer to the following code which can be checked in some IEDs like VSCode, Spyder, Anaconda and so on:

```
import sys # L1
from PyQt5.QtWidgets import QWidget, QApplication # L2


myapp = QApplication(sys.argv) # L3
mywindow = QWidget() # L4
mywindow.show() # L5
sys.exit(myapp.exec_()) # L6
```

Output:

The output can be seen in the following *Figure 1.4:*

Figure 1.4: *Output of Chap1_Example1.py*

Note: The preceding code is covered in Program Name: Chap1_Example1.py