|  |  |
| --- | --- |
| 1 | **QLabel**  A QLabel object acts as a placeholder to display non-editable text or image, or a movie of animated GIF. It can also be used as a mnemonic key for other widgets. |
| 2 | **QLineEdit**  QLineEdit object is the most commonly used input field. It provides a box in which one line of text can be entered. In order to enter multi-line text, QTextEdit object is required |
| 3 | **QPushButton**  In PyQt API, the QPushButton class object presents a button which when clicked can be programmed to invoke a certain function. |
| 4 | **QRadioButton**  A QRadioButton class object presents a selectable button with a text label. The user can select one of many options presented on the form. This class is derived from QAbstractButton class. |
| 5 | **QCheckBox**  A rectangular box before the text label appears when a QCheckBox object is added to the parent window. Just as QRadioButton, it is also a selectable button. |
| 6 | **QComboBox**  A QComboBox object presents a dropdown list of items to select from. It takes minimum screen space on the form required to display only the currently selected item. |
| 7 | **QSpinBox**  A QSpinBox object presents the user with a textbox which displays an integer with up/down button on its right. |
| 8 | **QSlider Widget & Signal**  QSlider class object presents the user with a groove over which a handle can be moved. It is a classic widget to control a bounded value. |
| 9 | **QMenuBar, QMenu & QAction**  A horizontal QMenuBar just below the title bar of a QMainWindow object is reserved for displaying QMenu objects. |
| 10 | **QToolBar**  A QToolBar widget is a movable panel consisting of text buttons, buttons with icons or other widgets. |
| 11 | **QInputDialog**  This is a preconfigured dialog with a text field and two buttons, OK and Cancel. The parent window collects the input in the text box after the user clicks on Ok button or presses Enter. |
| 12 | **QFontDialog**  Another commonly used dialog, a font selector widget is the visual appearance of QDialog class. Result of this dialog is a Qfont object, which can be consumed by the parent window. |
| 13 | **QFileDialog**  This widget is a file selector dialog. It enables the user to navigate through the file system and select a file to open or save. The dialog is invoked either through static functions or by calling exec\_() function on the dialog object. |
| 14 | **QTab**  If a form has too many fields to be displayed simultaneously, they can be arranged in different pages placed under each tab of a Tabbed Widget. The QTabWidget provides a tab bar and a page area. |
| 15 | **QStacked**  Functioning of QStackedWidget is similar to QTabWidget. It also helps in the efficient use of window’s client area. |
| 16 | **QSplitter**  If a form has too many fields to be displayed simultaneously, they can be arranged in different pages placed under each tab of a Tabbed Widget. The QTabWidget provides a tab bar and a page area. |
| 17 | **QDock**  A dockable window is a subwindow that can remain in floating state or can be attached to the main window at a specified position. Main window object of QMainWindow class has an area reserved for dockable windows. |
| 18 | **QStatusBar**  QMainWindow object reserves a horizontal bar at the bottom as the status bar. It is used to display either permanent or contextual status information. |
| 19 | **QList**  QListWidget class is an item-based interface to add or remove items from a list. Each item in the list is a QListWidgetItem object. ListWidget can be set to be multiselectable. |
| 20 | **QScrollBar**  A scrollbar control enables the user to access parts of the document that is outside the viewable area. It provides visual indicator to the current position. |
| 21 | **QCalendar**  QCalendar widget is a useful date picker control. It provides a month-based view. The user can select the date by the use of the mouse or the keyboard, the default being today’s date. |