# PyQt5常用模块

## PyQt5.QtCore

|  |  |  |  |
| --- | --- | --- | --- |
| QAbstractAnimation  QAbstractEventDispatcher  [QAbstractItemModel](#QAbstractTableModel)  QAbstractListModel  QAbstractNativeEventFilter  QAbstractProxyModel  QAbstractState  QAbstractTableModel  QAbstractTransition  QAnimationGroup  QBasicMutex  QBasicTimer  QBitArray  QBuffer  QByteArray  QByteArrayMatcher  QCalendar  QCborArray  QCborError  QCborMap  QCborParserError  QCborStreamReader  QCborStreamWriter  QCborStringResultByteArray  QCborStringResultString  QCborValue  QChildEvent  QCollator  QCollatorSortKey  QCommandLineOption  QCommandLineParser  QConcatenateTablesProxyModel  QCoreApplication  QCryptographicHash  QDataStream  QDate  QDateTime  QDeadlineTimer  QDir  QDirIterator  QDynamicPropertyChangeEvent  QEasingCurve  QElapsedTimer  QEnum/QFlag | QEvent  QEventLoop  QEventTransition  QFactoryInterface  QFile  QFileDevice  QFileInfo  QFileSelector  QFileSystemWatcher  QFinalState  QFutureInterfaceBase  QGenericArgument  QGenericReturnArgument  QHistoryState  QIdentityProxyModel  QIODevice  QItemSelection  QItemSelectionModel  QItemSelectionRange  QJsonArray  QJsonDocument  QJsonParseError  QJsonValue  QLibraryInfo  QLine  QLineF  QLocale  QLockFile  QMargins  QMarginsF  QMessageAuthenticationCode  QMessageLogContext  QMetaClassInfo  QMetaEnum  QMetaMethod  QMetaObject  QMetaProperty  QMimeData  QMimeDatabase  QMimeType  QModelIndex  QMutex  QMutexLocker  QObject | QOperatingSystemVersion  QParallelAnimationGroup  QPauseAnimation  QPersistentModelIndex  QPluginLoader  QPoint  QPointF  QProcess  QProcessEnvironment  QPropertyAnimation  QRandomGenerator  QRandomGenerator64  QReadLocker  QReadWriteLock  QRect  QRectF  QRecursiveMutex  QRegExp  QRegularExpression  QRegularExpressionMatch  QRegularExpressionMatchIterator  QResource  QRunnable  QSaveFile  QSemaphore  QSemaphoreReleaser  QSequentialAnimationGroup  QSettings  QSignalBlocker  QSignalMapper  QSignalTransition  QSize  QSizeF  QSocketDescriptor  QSocketNotifier  QSortFilterProxyModel  QStandardPaths  QState  QStateMachine  QStorageInfo  QStringListModel  QSysInfo  QSystemSemaphore  Qt | QTemporaryDir  QTemporaryFile  QTextBoundaryFinder  QTextCodec  QTextDecoder  QTextEncoder  QTextStream  QTextStreamManipulator  QThread  QThreadPool  QTime  QTimeLine  QTimer  QTimerEvent  QTimeZone  QTranslator  QTransposeProxyModel  QUrl  QUrlQuery  QUuid  QVariantAnimation  QVersionNumber  QWaitCondition  QWriteLocker  QXmlStreamAttribute  QXmlStreamAttributes  QXmlStreamEntityDeclaration  QXmlStreamEntityResolver  QXmlStreamNamespaceDeclaration  QXmlStreamNotationDeclaration  QXmlStreamReader  QXmlStreamWriter  ClassInfo  Connection  ConverterState  FromBase64Result  OffsetData  Property  Signal  SignalEvent  Slot  TimerInfo  WrappedEvent  YearMonthDay |

【详细描述】提供核心的非GUI功能的类，包含QFile，QDir，QTimer等非界面组件类；

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

## PyQt5. QtGui

|  |  |  |  |
| --- | --- | --- | --- |
| QAbstractOpenGLFunctions  QAbstractTextDocumentLayout  QAccessible  QAccessibleEditableTextInterface  QAccessibleEvent  QAccessibleInterface  QAccessibleObject  QAccessibleStateChangeEvent  QAccessibleTableCellInterface  QAccessibleTableModelChangeEvent  QAccessibleTextCursorEvent  QAccessibleTextInsertEvent  QAccessibleTextInterface  QAccessibleTextRemoveEvent  QAccessibleTextSelectionEvent  QAccessibleTextUpdateEvent  QAccessibleValueChangeEvent  QAccessibleValueInterface  QActionEvent  QBackingStore  QBitmap  QBrush  QClipboard  QCloseEvent  QColor  QColorConstants  QColorSpace  QConicalGradient  QContextMenuEvent  QCursor  QDesktopServices  QDoubleValidator  QDrag  QDragEnterEvent  QDragLeaveEvent  QDragMoveEvent  QDropEvent  QEnterEvent  QExposeEvent  QFileOpenEvent  QFocusEvent  QFont  QFontDatabase  QFontInfo  QFontMetrics  QFontMetricsF  QGradient  QGuiApplication  QHelpEvent  QHideEvent  QHoverEvent | QIcon  QIconDragEvent  QIconEngine  QImage  QImageIOHandler  QImageReader  QImageWriter  QInputEvent  QInputMethod  QInputMethodEvent  QInputMethodQueryEvent  QIntValidator  QKeyEvent  QKeySequence  QLinearGradient  QMatrix  QMatrix2x2  QMatrix2x3  QMatrix2x4  QMatrix3x2  QMatrix3x3  QMatrix3x4  QMatrix4x2  QMatrix4x3  QMatrix4x4  QMouseEvent  QMoveEvent  QMovie  QNativeGestureEvent  QOffscreenSurface  QOpenGLBuffer  QOpenGLContext  QOpenGLContextGroup  QOpenGLDebugLogger  QOpenGLDebugMessage  QOpenGLExtraFunctions  QOpenGLFramebufferObject  QOpenGLFramebufferObjectFormat  QOpenGLFunctions  QOpenGLPixelTransferOptions  QOpenGLShader  QOpenGLShaderProgram  QOpenGLTexture  QOpenGLTextureBlitter  QOpenGLTimeMonitor  QOpenGLTimerQuery  QOpenGLVersionProfile  QOpenGLVertexArrayObject  QOpenGLWindow  QPagedPaintDevice  QPageLayout | QPageSize  QPaintDevice  QPaintDeviceWindow  QPaintEngine  QPaintEngineState  QPainter  QPainterPath  QPainterPathStroker  QPaintEvent  QPalette  QPdfWriter  QPen  QPicture  QPictureIO  QPixelFormat  QPixmap  QPixmapCache  QPointingDeviceUniqueId  QPolygon  QPolygonF  QQuaternion  QRadialGradient  QRasterWindow  QRawFont  QRegExpValidator  QRegion  QRegularExpressionValidator  QResizeEvent  QScreen  QScrollEvent  QScrollPrepareEvent  QSessionManager  QShortcutEvent  QShowEvent  QStandardItem  QStandardItemModel  QStaticText  QStatusTipEvent  QStyleHints  QSurface  QSurfaceFormat  QSyntaxHighlighter  Qt  QTabletEvent  QTextBlock  QTextBlockFormat  QTextBlockGroup  QTextBlockUserData  QTextCharFormat  QTextCursor  QTextDocument | QTextDocumentFragment  QTextDocumentWriter  QTextFormat  QTextFragment  QTextFrame  QTextFrameFormat  QTextImageFormat  QTextInlineObject  QTextItem  QTextLayout  QTextLength  QTextLine  QTextList  QTextListFormat  QTextObject  QTextObjectInterface  QTextOption  QTextTable  QTextTableCell  QTextTableCellFormat  QTextTableFormat  QToolBarChangeEvent  QTouchDevice  QTouchEvent  QTransform  QValidator  QVector2D  QVector3D  QVector4D  QWhatsThisClickedEvent  QWheelEvent  QWindow  QWindowStateChangeEvent  Attribute  AvailableSizesArgument  Binder  Element  FormatRange  iterator  iterator  Key  Margins  PaintContext  PixmapFragment  Selection  State  Svg  Tab  TouchPoint |

【详细描述】QtGui：提供GUI设计中用于窗口系统集成，事件处理，绘图等功能的类，包含QIcon，Qfont，Qpixmap，QCloseEvent，QPalette，QPainter等GUI底层实现类；

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

## PyQt5.QtWidgets

|  |  |  |  |
| --- | --- | --- | --- |
| [QAbstractButton](#QAbstractButton)  QAbstractGraphicsShapeItem  QAbstractItemDelegate  QAbstractItemView  QAbstractScrollArea  QAbstractSlider  QAbstractSpinBox  QAccessibleWidget  [QAction](#QAction)  QActionGroup  [QApplication](#QApplication)  QBoxLayout  QButtonGroup  QCalendarWidget  QCheckBox  QColorDialog  QColormap  QColumnView  QComboBox  QCommandLinkButton  QCommonStyle  QCompleter  QDataWidgetMapper  QDateEdit  QDateTimeEdit  QDesktopWidget  QDial  QDialog  QDialogButtonBox  QDirModel  QDockWidget  QDoubleSpinBox  QErrorMessage  QFileDialog  QFileIconProvider  QFileSystemModel  QFocusFrame  QFontComboBox  QFontDialog  QFormLayout  QFrame  QGesture  QGestureEvent  QGestureRecognizer  QGraphicsAnchor  QGraphicsAnchorLayout  QGraphicsBlurEffect  QGraphicsColorizeEffect  QGraphicsDropShadowEffect  QGraphicsEffect | QGraphicsEllipseItem  QGraphicsGridLayout  QGraphicsItem  QGraphicsItemAnimation  QGraphicsItemGroup  QGraphicsLayout  QGraphicsLayoutItem  QGraphicsLinearLayout  QGraphicsLineItem  QGraphicsObject  QGraphicsOpacityEffect  QGraphicsPathItem  QGraphicsPixmapItem  QGraphicsPolygonItem  QGraphicsProxyWidget  QGraphicsRectItem  QGraphicsRotation  QGraphicsScale  QGraphicsScene  QGraphicsSceneContextMenuEvent  QGraphicsSceneDragDropEvent  QGraphicsSceneEvent  QGraphicsSceneHelpEvent  QGraphicsSceneHoverEvent  QGraphicsSceneMouseEvent  QGraphicsSceneMoveEvent  QGraphicsSceneResizeEvent  QGraphicsSceneWheelEvent  QGraphicsSimpleTextItem  QGraphicsTextItem  QGraphicsTransform  QGraphicsView  QGraphicsWidget  QGridLayout  QGroupBox  QHBoxLayout  QHeaderView  QInputDialog  QItemDelegate  QItemEditorCreatorBase  QItemEditorFactory  QKeyEventTransition  QKeySequenceEdit  QLabel  QLayout  QLayoutItem  QLCDNumber  QLineEdit  QListView  QListWidget | QListWidgetItem  QMainWindow  QMdiArea  QMdiSubWindow  QMenu  QMenuBar  QMessageBox  QMouseEventTransition  QOpenGLWidget  QPanGesture  QPinchGesture  QPlainTextDocumentLayout  QPlainTextEdit  QProgressBar  QProgressDialog  QProxyStyle  QPushButton  QRadioButton  QRubberBand  QScrollArea  QScrollBar  QScroller  QScrollerProperties  QShortcut  QSizeGrip  QSizePolicy  QSlider  QSpacerItem  QSpinBox  QSplashScreen  QSplitter  QSplitterHandle  QStackedLayout  QStackedWidget  QStatusBar  QStyle  QStyledItemDelegate  QStyleFactory  QStyleHintReturn  QStyleHintReturnMask  QStyleHintReturnVariant  QStyleOption  QStyleOptionButton  QStyleOptionComboBox  QStyleOptionComplex  QStyleOptionDockWidget  QStyleOptionFocusRect  QStyleOptionFrame  QStyleOptionGraphicsItem  QStyleOptionGroupBox | QStyleOptionHeader  QStyleOptionMenuItem  QStyleOptionProgressBar  QStyleOptionRubberBand  QStyleOptionSizeGrip  QStyleOptionSlider  QStyleOptionSpinBox  QStyleOptionTab  QStyleOptionTabBarBase  QStyleOptionTabWidgetFrame  QStyleOptionTitleBar  QStyleOptionToolBar  QStyleOptionToolBox  QStyleOptionToolButton  QStyleOptionViewItem  QStylePainter  QSwipeGesture  QSystemTrayIcon  QTabBar  QTableView  QTableWidget  QTableWidgetItem  QTableWidgetSelectionRange  QTabWidget  QTapAndHoldGesture  QTapGesture  QTextBrowser  QTextEdit  QTileRules  QTimeEdit  QToolBar  QToolBox  QToolButton  QToolTip  QTreeView  QTreeWidget  QTreeWidgetItem  QTreeWidgetItemIterator  QUndoCommand  QUndoGroup  QUndoStack  QUndoView  QVBoxLayout  QWhatsThis  QWidget  QWidgetAction  QWidgetItem  QWizard  QWizardPage  ExtraSelection |

【详细描述】提供GUI设计中所有窗体显示的类，包含各种窗体，标准对话框，按钮，文本框等组件，包含QMainWindow，QWidget，QDiglog等窗口，QColorDialog，QFileDialog等标准对话框，QRadioButton，QSpinBox，QMenu等界面组件；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## PyQt5.QtSql

|  |  |  |  |
| --- | --- | --- | --- |
| QSql  QSqlDatabase  QSqlDriver  QSqlDriverCreatorBase | QSqlError  QSqlField  QSqlIndex  QSqlQuery | QSqlQueryModel  QSqlRecord  QSqlRelation  QSqlRelationalDelegate | QSqlRelationalTableModel  QSqlResult  QSqlTableModel |

【详细描述】QtSql：提供SQL数据库驱动，数据查询和操作的类，包含QSqlDatabase，QSqlQuery，QSqlRecord等；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## PyQt5.QtMultimedia

【详细描述】提供音频，视频，摄像头操作的类，包含QCamera，QAudioInput，QMediaPlayer等

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## PyQt5.QtMultimeidaWidget

详细描述提供多媒体窗体显示的类，包含QCameraViewfinder，QVideoWidget等

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

定义：

1. [Fun]：Functions；

2. [Sl]：Slots；

3. [Sig]：Signals；

4. [StF]：Static functions

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

# PyQt5常用类（Class）

## QCoreApplication

【继承关系】QtCore.QCoreApplication

【继承者】QGuiApplication, QApplication

|  |  |  |  |
| --- | --- | --- | --- |
| aboutToQuit  addLibraryPath  applicationDirPath  applicationFilePath  applicationName  applicationPid  applicationVersion  arguments  childEvent  closingDown  connectNotify  customEvent  disconnectNotify  event | eventDispatcher  exec  exec\_  exit  flush  hasPendingEvents  installNativeEventFilter  installTranslator  instance  isQuitLockEnabled  isSetuidAllowed  isSignalConnected  libraryPaths  notify | organizationDomain  organizationName  postEvent  processEvents  quit  receivers  removeLibraryPath  removeNativeEventFilter  removePostedEvents  removeTranslator  sender  senderSignalIndex  sendEvent  sendPostedEvents | setApplicationName  setApplicationVersion  setAttribute  setEventDispatcher  setLibraryPaths  setOrganizationDomain  setOrganizationName  setQuitLockEnabled  setSetuidAllowed  startingUp  testAttribute  timerEvent  translate |

【详细描述】

processEvents：使应用程序处理完事件队列里所有未处理的事件，函数原型processEvents(flags = QEventLoop.allEvents)，其中falgas是枚举值，QEventLoop.ProcessEventsFlag对象，表示处理的方式【1. AllEvents，处理所有事件；ExcludeUserInputEvent，排除用户输入事件，如键盘和鼠标的事件；ExcludeSocketNotifiers，排除网络socket的通知事件；WaitForMoreEvents，如果没有未处理的事件，等待更多事件】

translate：用于实现程序的国际化(或者说多语言翻译)的函数，参数1. context 上下文，一般就是需要翻译的字符串所在的类的名字；2. sourceText 需要翻译的字符串；3. disambiguation 用于消除歧义。

## QApplication

【继承关系】QtWidgets.QApplication -> QtGui.QGuiApplication –> QtCore.QCoreApplication

|  |  |  |  |
| --- | --- | --- | --- |
| aboutQt  activeModalWidget  activePopupWidget  activeWindow  alert  allWidgets  autoSipEnabled  beep  childEvent  closeAllWindows  colorSpec  connectNotify  cursorFlashTime  customEvent  desktop | disconnectNotify  doubleClickInterval  event  exec\_  focusChanged  focusWidget  font  fontMetrics  globalStrut  isEffectEnabled  isSignalConnected  keyboardInputInterval  notify  palette  receivers | sender  senderSignalIndex  setActiveWindow  setAutoSipEnabled  setColorSpec  setCursorFlashTime  setDoubleClickInterval  setEffectEnabled  setFont  setGlobalStrut  setKeyboardInputInterval  setPalette  setStartDragDistance  setStartDragTime  setStyle | setStyleSheet  setWheelScrollLines  setWindowIcon  startDragDistance  startDragTime  style  styleSheet  timerEvent  topLevelAt  topLevelWidgets  wheelScrollLines  widgetAt  windowIcon |

【详细描述】用于管理GUI程序的控制流和主要设置，其包含窗口系统和其他来源处理过和发送过的主事件循环，它也处理应用程序的初始化和收尾工作，并提供对话管理（可以对系统和应用的大部分设置项进行设置）。

对于用Qt写的任何一个GUI应用，不管这个应用有没有窗口或多少个窗口，有且只有一个QApplication对象。而对于用Qt写的非GUI应用，则有且只有一个QCoreApplication对象，并且这个应用不依赖QtGui库。

QApplication的主要功能是：

1. 使用所在桌面的参数例如调色板[palette()]、字体[font()]、双击间隔[doubleClickInterval()]等信息初始化应用程序。当我们改变桌面的一些设置时，可以让应用程序的这些设置保持一致。

2. 处理事件，从窗口系统接收事件并派发到相应的Widget，使用sendEvent()和postEvent()函数可以派发事件。

3. 处理命令行参数，设置内部状态。

4、定义GUI外观，外观由QStyle对象包装，运行时通过setStyle()函数进行设置。。

5、设置程序的颜色分配策略，详见setColorSpec()函数说明。

6、通过translate()函数可以设置本地化的字符串。

7. 提供了一些有用的对象，例如屏幕信息类[desktop()]和剪切板类[clipboard()]。

8. QApplication包含所有窗口的信息，可以知道程序在屏幕上的位置（widgetAt()），顶层窗口列表（topLevelWidgets()），关闭所有窗口（closeAllWindows()）等等。

9. QApplication能够管理鼠标光标样式，参照setOverrideCursor()。

10. QApplication提供全面的与用户交互界面。当用户注销时，可以正常关闭；如果不能终止程序，它能取消关机进程；甚至可以保存程序全部状态。参看 isSessionRestored()，sessionId() ，commitData()，saveState()

从上面可以看出，QApplication作了许多初始化工作，因此在任何其它的UI对象创建之前必须先创建QApplication对象，而且还可以通过命令行参数设置一些内部状态。

创建QApplication类：app = QApplication(sys.argv)。sys.argv是参数变量[argument variable]的简写形式，即CMD命令行。这个变量其实是一个命令行参数列表，argv[0]为被调用的程序路径，argv[1]和以后就是传入的数据了，在CMD中显示空格分隔的命令。

【函数介绍】

aboutQt：显示一个关于Qt的简单消息框，该消息包含应用程序正在使用的Qt的版本号。

activeModalWidget：返回激活的模式窗口部件，模式窗口部件是特殊的顶级窗口部件，它是QDialog的子类。

activePopupWidget：返回激活弹出窗口部件。弹出窗口部件是特殊的顶级窗口部件，它设置了窗口部件标记WType\_Popup，例如QPopupMenu窗口部件。当应用程序打开一个弹出窗口部件，所有的事件都被发送给弹出窗口部件。在弹出窗口部件被关闭之前，普通窗口部件和模式对话框都不能被访问。

activeWindow：返回拥有键盘输入焦点的应用程序顶级窗口，如果没有应用程序窗口拥有焦点，则返回0。

desktop: 返回桌面窗口部件（也调用根窗口），返回QDesktopWidget对象 [若使用方法width()，height()分别返回返回桌面宽度，高度]。

allWidgets：返回应用程序中所有窗口部件的列表，如果没有窗口部件，返回空列表。

beep：返回提示音。

globalStrut：保留用户可以与之交互的任何GUI元素应具有的最小大小，返回QSize对象。

closeAllWindows：关闭所有顶级窗口，它可以被和文件菜单中的“Quit”条目连接，实现关闭窗口的动作：

colorSpec：返回颜色规格。

cursorFlashTime：返回文本关标闪烁（闪亮）的时间，以毫秒为单位。闪烁时间是显示、转化和恢复脱字符显示所使用的时间。

doubleClickInterval：返回双击的最大持续时间。

exec：进入主事件循环并且等待，直到exit()被调用或者主窗口部件被销毁，并且返回值被设置为0（如果通过quit()，exit()被调用，返回值为0）。

exit：告诉应用程序退出，并返回0。

quit：关闭窗口

focusWidget：返回拥有键盘输入焦点的应用程序窗口部件，如果应用程序中没有窗口部件拥有焦点，则返回0。

font：返回窗口部件w的默认字体，或者如果w为0，返回默认应用程序字体。

fontMetrics：返回应用程序字体的显示（屏幕）字体规格。

globalStrut：返回应用程序的全局strut，strut是任何用户可以交互的图形交互界面元素的最小维数的大小对象。例如，没有按钮应该被重新定义大小小于全局strut大小。

palette：返回默认应用程序调色板的指针。

setColorSpec：设置应用程序的颜色策略。颜色策略控制着运行应用程序的颜色分配，故必须在创建QApplication对象之前进行设置；[QApplication.NormalColor，默认颜色分配策略；QApplication.CustomColor，使用少量的自定义颜色。QApplication.ManyColor，使用非常大量的颜色]。

instance：获取QApplication的指针，基本与全局指针QtWidgets.qApp等价[返回应用程序对象]。

setCursorFlashTime：设置文本光标闪烁（闪亮）时间为msecs毫秒。闪烁时间是显示、转化和恢复脱字符显示所使用的时间。

setDoubleClickInterval：设置区别一个双击和两次单击的时间限度，单位为ms。

setFont：改变应用程序默认字体为Qfont，参数为QFont，格式：QFont('华文楷体', 10)。

setGlobalStrut：设置应用程序全局的strut，参数为QSize，格式：QSize(int, int)。

setPalette：设置应用程序全局的Qpalette对象；

startDragDistance：如果应用程序支持拖放并且拖动应该在鼠标点击后移动一定距离才开始，该方法返回这个距离；

setStartDragDistance：设置这个距离；

setStartDragTime：如果应用程序支持拖放并且拖动应该在鼠标点击后一定时间才开始，该方法返回这个延迟时间（以ms为单位）；

setStyle：设置应用程序的窗口风格。例如：setStyle(QStyleFactory.create(“WindowsXP”))

style：返回应用程序的窗口风格，可以通过objectName() 获取，获取当前平台支持的原有QStyle样式可用QStyleFactory.keys()，一般为['windowsvista', 'Windows', 'Fusion']

setStyleSheet：设置按钮或QWidget的外观。例如：setStyleSheet("QTableWidget {background-color: black;border:20px solid #014F84}" "QTableWidget::item{border:1px solid #014F84}")

setWindowIcon：设置窗口图标，输入参数为QIcon；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QObject

【继承者】...

|  |  |  |  |
| --- | --- | --- | --- |
| blockSignals  childEvent  children  connectNotify  customEvent  deleteLater  destroyed  disconnect  disconnectNotify  dumpObje  ctInfo | dumpObjectTree  dynamicPropertyNames  event  eventFilter  findChild  findChildren  inherits  installEventFilter  isSignalConnected  isWidgetType | isWindowType  killTimer  metaObject  moveToThread  objectName  objectNameChanged  parent  property  pyqtConfigure  receivers | removeEventFilter  sender  senderSignalIndex  setObjectName  setParent  setProperty  signalsBlocked  startTimer  thread  timerEvent  tr |

【详细描述】QObject类是所有Qt对象的基类，也是Qt对象模型的核心，这个模型的主要特征为信号与槽，信号与槽是一个非常重要的机制，用于对象间的通信。连接信号与槽使用connect函数，断开连接使用disconnect函数，blockSignals()函数可以临时屏蔽信号，避免循环不断的通知，另外，protected级别的connectNotify和disconnectNotify函数还可用于跟踪信号与槽的连接关系。

QObject对象是一种对象树的形式，当创建一个QObject对象并指定另外一个对象为parent时，这个对象将自动把自己添加到parent的children()列表，这样，parent拥有这个对象的所有权，parent销毁时自动销毁children，可以通过名字（objectName）查找parent拥有的children，使用findChild()或findChildren()函数。

每个对象都有一个objectName()函数，其类名可以通过metaObject()函数获得（QMetaObject.className()），你也可以在QObject类的树状继承结构中，查看一个对象的类是否继承自另一个类，使用inherits()函数。

一个对象销毁时，发送destroyed信号，可以捕获这个信号以避免对这个对象的无效引用。

QObject对象可以通过event()函数接收事件并过滤其它对象的事件（相关函数为installEventFilter()和eventFilter()），同时通过重写childEvent()函数还可以捕获children的事件。

此外，QObject还提供了基本的timer功能的支持，具体关于timer的高级功能可参照QTimer类。

\* Q\_OBJECT是一个很重要的宏，QObject对象实现了信号、槽、属性时，这个宏是必需的，元对象编译器moc（Meta Object Compiler）需要用到这个宏，建议QObject的所有子类不管有没有用到信号、槽、属性都使用这个宏，否则有可能出现诡异的事情。moc相关说明可参照Qt文档Using the Meta-Object Compiler (moc)。

\* 所有的Qt Widget都继承自QObject类，判断一个QObject对象是否为Widget可以使用isWidgetType()函数进行判断，这个函数比qobject\_cast<QWidget \*>(obj)或obj->inherits("QWidget")等方法更快。

Object的某些函数，如children()，返回类型为QObjectList，这个是用typedef定义的QList<QObject\*>。

【常用函数】

setObjectName：设置对象名称，参数为str；objectName：获取对象的名称；

setProperty(str, xx )：添加对象动态属性与值；property("property\_name")：获得对象某一属性对应的值, 将返回value；

setParent(obj1)：设置父对象，将obj1设置为obj2的父对象

parent：获取父对象；

children：获取所有直接子对象；

findChild(参数1, 参数2, 参数3)：获取某一个指定名称和类型的子对象

findChildren(参数1, 参数2, 参数3)：获取某多个指定名称和类型的子对象

blockSignals(True/False)：临时（取消）阻止指定控件所有的信号与槽的连，True为临时阻断，False为重新连接

signalsBlock：查看信号是否被阻断

inherits(‘父类’)：判断一个对象是否继承（直接继承或间接继承）于某个类

deleteLater：对象删除

mro：查看对象的整个继承链条（obj. \_\_bases\_\_ # 查看对象的父类）；

installEventFilter：实现事件过滤器的安装，参数为窗体对象；

eventFilter：对被监测的对象及其事件进行处理，参数1.watched->QObject对象，表示事件发生的对象；even是QEvent对象，用于表示事件类型event.type()对象，eventFilter函数代码最后一句必须是return super().eventFilter(watched,event)；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMetaObject

【继承关系】QtCore.QMetaObject

|  |  |  |  |
| --- | --- | --- | --- |
| checkConnectArgs  classInfo  classInfoCount  classInfoOffset  className  connectSlotsByName  constructor  constructorCount | enumerator  enumeratorCount  enumeratorOffset  indexOfClassInfo  indexOfConstructor  indexOfEnumerator  indexOfMethod  indexOfProperty | indexOfSignal  indexOfSlot  inherits  invokeMethod  method  methodCount  methodOffset  normalizedSignature | normalizedType  property  propertyCount  propertyOffset  superClass  userProperty |

【详细描述】元对metaObject

connectSlotsByName：递归搜索给定对象的所有子对象，并将来自这些子对象的匹配信号连接到以下形式的对象槽，参数1，子对象名称；对象槽on\_<object name>\_<signal name>(<signal parameters>)

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QWidget

【继承关系】QtWidgets.QWidget -> QtCore.QObject/ QtGui.QPaintDevice

|  |  |  |  |
| --- | --- | --- | --- |
| acceptDrops  accessibleDescription  accessibleName  actionEvent  actions  activateWindow  addAction  addActions  adjustSize  autoFillBackground  backgroundRole  baseSize  changeEvent  childAt  childEvent  childrenRect  childrenRegion  clearFocus  clearMask  close  closeEvent  connectNotify  contentsMargins  contentsRect  contextMenuEvent  contextMenuPolicy  create  createWindowContainer  cursor  customContextMenuRequested  customEvent  destroy  devType  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  effectiveWinId  ensurePolished  enterEvent  event  find  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPolicy  focusPreviousChild  focusProxy  focusWidget  font  fontInfo  fontMetrics  foregroundRole  frameGeometry  frameSize  geometry  getContentsMargins  grab  grabGesture  grabKeyboard  grabMouse  grabShortcut  graphicsEffect  graphicsProxyWidget  hasFocus | hasHeightForWidth  hasMouseTracking  hasTabletTracking  height  heightForWidth  hide  hideEvent  initPainter  inputMethodEvent  inputMethodHints  inputMethodQuery  insertAction  insertActions  isActiveWindow  isAncestorOf  isEnabled  isEnabledTo  isFullScreen  isHidden  isLeftToRight  isMaximized  isMinimized  isModal  isRightToLeft  isSignalConnected  isVisible  isVisibleTo  isWindow  isWindowModified  keyboardGrabber  keyPressEvent  keyReleaseEvent  layout  layoutDirection  leaveEvent  locale  lower  mapFrom  mapFromGlobal  mapFromParent  mapTo  mapToGlobal  mapToParent  mask  maximumHeight  maximumSize  maximumWidth  metric  minimumHeight  minimumSize  minimumSizeHint  minimumWidth  mouseDoubleClickEvent  mouseGrabber  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  move  moveEvent  nativeEvent  nativeParentWidget  nextInFocusChain  normalGeometry  overrideWindowFlags  overrideWindowState  paintEngine  paintEvent | palette  parentWidget  pos  previousInFocusChain  raise\_  receivers  rect  releaseKeyboard  releaseMouse  releaseShortcut  removeAction  render  repaint  resize  resizeEvent  restoreGeometry  saveGeometry  screen  scroll  sender  senderSignalIndex  setAcceptDrops  setAccessibleDescription  setAccessibleName  setAttribute  setAutoFillBackground  setBackgroundRole  setBaseSize  setContentsMargins  setContextMenuPolicy  setCursor  setDisabled  setEnabled  setFixedHeight  setFixedSize  setFixedWidth  setFocus  setFocusPolicy  setFocusProxy  setFont  setForegroundRole  setGeometry  setGraphicsEffect  setHidden  setInputMethodHints  setLayout  setLayoutDirection  setLocale  setMask  setMaximumHeight  setMaximumSize  setMaximumWidth  setMinimumHeight  setMinimumSize  setMinimumWidth  setMouseTracking  setPalette  setParent  setShortcutAutoRepeat  setShortcutEnabled  setSizeIncrement  setSizePolicy  setStatusTip  setStyle  setStyleSheet  setTabletTracking  setTabOrder | setToolTip  setToolTipDuration  setUpdatesEnabled  setVisible  setWhatsThis  setWindowFilePath  setWindowFlag  setWindowFlags  setWindowIcon  setWindowIconText  setWindowModality  setWindowModified  setWindowOpacity  setWindowRole  setWindowState  setWindowTitle  sharedPainter  show  showEvent  showFullScreen  showMaximized  showMinimized  showNormal  size  sizeHint  sizeIncrement  sizePolicy  stackUnder  statusTip  style  styleSheet  tabletEvent  testAttribute  timerEvent  toolTip  toolTipDuration  underMouse  ungrabGesture  unsetCursor  unsetLayoutDirection  unsetLocale  update  updateGeometry  updateMicroFocus  updatesEnabled  visibleRegion  whatsThis  wheelEvent  width  window  windowFilePath  windowFlags  windowHandle  windowIcon  windowIconChanged  windowIconText  windowIconTextChanged  windowModality  windowOpacity  windowRole  windowState  windowTitle  windowTitleChanged  windowType  winId  x  y |

【详细描述】窗口控件，所有用户界面对象的基类，

setGeometry：设置控件在windows窗口中的位置及大小，参数为QRect对象；geometry：获取QWidget窗口的大小，返回screen.width, screen.high对象，其成员函数x, y获得客户区左上角的坐标，width, height获得客户区的宽度和高度；

frameGeometry：获取窗口的高度、宽度、顶点和左边位置，返回QtCore.QRect对象，其成员函数x,y获得窗口左上角的坐标，width,height获得整个窗口的宽度和高度；

move：将QWidget窗口移动到屏幕的固定坐标，参数为screen.width, screen.high，或者QPoint对象；

resize：设置QWidget窗口的大小，参数为int, 分别表示宽度和高度，screen.width, screen.high；size：获取客户区的大小；

width, height：获取控件的宽度和高度；

pos：获取QWidget窗口左上角坐标【相对坐标】；

mapToGlobal：将当前控件的相对位置转换为屏幕绝对位置，参数为QPoint对象；

mapFromGlobal：将当前控件的绝对位置坐标对应到控件的相对位置的位置；

setFiexedWidth/setFiexedheight：将控件的宽度和高度设置为某个固定值；

setSizePolicy：设置尺寸策略，参数为QSizePolicy对象；

x,y：获得窗口左上角的坐标。width,height：获得客户区的宽度和高度

setWindowTitle：设置窗口控件的标题。

setToolTip：创建气泡提示窗口；

setAutoFillBackground：设置自动填充背景，参数为Ture/False；

setWindowModality：设置窗口模态，参数为枚举值，参见Qt.WindowModality对象；

sender：当某一个Object发送一个signal的时候，它就是一个sender，系统会记录下当前是谁触发这个signal的，所以你在对应的slot里就可以通过 sender()得到当前是谁触发了你的

setContextMenuPolicy：设置允许右键产生菜单，其参数为枚举值，参见Qt.ContextMenuPolicy对象；

setContentsMargins：设置四个边的边距，参数为int，int，int，int分别表示；

setWindowFlags：改变窗口的类型，参数为枚举值，参见Qt.WindowType对象；

setWindowState：使窗口处于最小化，最大化等状态，参数Qt.WindowState对象；

setAttribute：用于设置窗体的一些属性，其参数1. 枚举值，Qt.WidgetAttribute对象；2. bool；

setWindowOpacity：用于设置窗口的透明度，参数level,从1.0（完全不透明）至0.0（完全透明）之间的浮点数；

setStyle：设置应用程序的窗口风格。例如：setStyle(QStyleFactory.create(“WindowsXP”))；style：返回应用程序的窗口风格，可以通过objectName() 获取，获取当前平台支持的原有QStyle样式可用QStyleFactory.keys()，一般为['windowsvista', 'Windows', 'Fusion']

setFixedSize：固定Widget窗口大小为参数设定值self.width(), self.height()，并禁止调整窗口大小

event：一个界面组件产生的事件首先会发送给其event函数做处理，如果event函数不做任何处理，就自动调用时间对应的默认处理函数，因此event函数可以用于一些事件屏蔽和预处理工作，其参数event是QEvent类型，可用event.type返回事件的具体类型，函数代码最后一句必须是return super().event(event)

paintEvent：当界面需要重新绘制时触发该事件；

mousePressEvent：鼠标按键按下时触发该事件；mouseReleaseEvent：鼠标按键释放时触发该事件；mouseMoveEvent：鼠标移动时触发该事件；mouseDoubleClickEvent：鼠标双击时触发该事件；

keyPressEvent：键盘按键按下时触发该事件；keyReleaseEvent：键盘按键释放时触发该事件；

closeEvent：一个窗体关闭时触发该事件；showEvent：一个窗体显示时触发该事件；hideEvent：一个窗体隐藏时触发该事件；

resizeEvent：组件改变大小时触发该事件；focusInEvent：当一个组件获取键盘焦点时触发；focusOutEvent：当一个组件失去键盘焦点时触发；

setAcceptDrops：设置B可以接受拖动的对象A，参数为Bool，如果设置为False，则组件不接受放置操作；如果设置为True，则会有默认的操作，并可以重新实现dragEnterEvent()和dropEvent()等事件函数实现特定的操作；

acceptDrops：返回当前对象是否可以接受拖动的判断；

enterEvent：当鼠标进入组件的屏幕空间时触发该事件；leaveEvent：当鼠标离开组件的屏幕空间时触发该事件；

dragEnterEvent：当执行一个拖拽空间操作，且鼠标指针进入该控件时，触发该事件。在该事件中可以获得被操作的窗口控件，还可以有田间的接受或拒绝该拖拽操作；

dragMoveEvent：在执行拖拽操作是会触发该事件；

dragLeaveEvent：当执行一个拖拽控件操作，并且鼠标指针离开该控件是，触发该事件；

dropEvent：当拖拽操作在目标控件上被释放时，触发该事件；

customContextMenuRequested：这个信号是QWidget唯一与右键菜单有关的信号(也是自有的唯一信号), 同时也是很容易被忽略的信号(signal)，如：listWidget.customContextMenuRequested[QtCore.QPoint].connect()

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMainWindow

【继承关系】QtWidgets.QMainWindow -> QtWidgets.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addDockWidget  addToolBar  addToolBarBreak  centralWidget  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  corner  create  createPopupMenu  customEvent  destroy  disconnectNotify  dockOptions  dockWidgetArea  documentMode  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event | focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  iconSize  iconSizeChanged  initPainter  inputMethodEvent  insertToolBar  insertToolBarBreak  isAnimated  isDockNestingEnabled  isSeparator  isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  menuBar  menuWidget  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent | mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  removeDockWidget  removeToolBar  removeToolBarBreak  resizeDocks  resizeEvent  restoreDockWidget  restoreState  saveState  sender  senderSignalIndex  setAnimated  setCentralWidget  setCorner  setDockNestingEnabled  setDockOptions  setDocumentMode  时  setMenuBar  setMenuWidget  setStatusBar | setTabPosition  setTabShape  setToolButtonStyle  setUnifiedTitleAndToolBarOnMac  sharedPainter  showEvent  splitDockWidget  statusBar  tabifiedDockWidgetActivated  tabifiedDockWidgets  tabifyDockWidget  tabletEvent  tabPosition  tabShape  takeCentralWidget  timerEvent  toolBarArea  toolBarBreak  toolButtonStyle  toolButtonStyleChanged  unifiedTitleAndToolBarOnMac  updateMicroFocus  wheelEvent |

【详细描述】QMainWindow主窗口，可以包含菜单栏，工具栏，状态栏，标题栏；（QWidget, QDialog也可以用来创建窗口，QDialog是对话框窗口的基类，主要用来执行短期任务或者与用户进行互动，其没有菜单栏，工具栏，状态栏等）

QMainWindow相关的属性：1. iconSize属性，用于设置主窗口中工具栏的图标尺寸；2. animated属性，用于设置在操作可浮动部件和工具栏时是否设置动画；3. toolButtonStyle属性，用于设置主窗口工具栏按钮的样式设置，其参数为枚举值，Qt.ToolButtonStyl对象；4. documentMode属性，用于设置当前主窗口是否启用文档模式；5. tabShape属性，用于设置主窗口标签部件（QTabWidget）中的标签的形状，其参数为枚举值，QTabWidget.TabShape对象；6. unifiedTitleAndToolBarOnMac属性，用于在mac操作系统上是否使用统一的标题和工具栏外观；7. dockOptions属性，用于设置主窗口对浮动部件停靠的反应，其参数为枚举值，QMainWindow.DockOption对象；8. dockNestingEnabled，用于设置主窗口的浮动部件（dock widget）是否允许嵌套的一个属性

addToolBar：添加工具栏，该控件是由文本按钮、图标或者其他小控件按钮组成的可移动面板，通常位于菜单栏下方，作为工具栏使用；

addDockWidget：用于将对应停靠窗（DockWidget）增加到主窗口的指定区域，

setAnimated：设置在操作可浮动部件和工具栏时是否设置动画，参数为bool；isAnimated：返回是否设置动画；

setDockNestingEnabled：设置是否可以嵌套停靠，参数为bool；

setDockOptions：设置DockOptions在QMainWindow的停靠行为，参数为枚举值，QMainWindow.DockOptions对象；

setTabShape：设置QTableWidge对象的标签形状，参数为枚举值，QTabWidget.TabShape对象；

createPopupMenu：用于创建弹出菜单；

manuBar： 返回主窗口的菜单栏，即QMenuBar对象。QMenu对象提供了一个可以添加菜单栏的控件，也可以用于创建上下文菜单和弹出菜单选项；

centralWidget：返回窗口中心的一个控件，未设置时返回NULL

setCentralWidget：返回窗口中心的控件；

statusBar：获得状态栏，调用状态栏对象的showMessage(“XXXX”, msecs = int(0000) )方法，显示状态栏信息，并悬停0000毫秒，0为永久显示；

setStatusBar：设置状态栏；

setToolButtonStyle：设置ToolButton按钮显示文字、图标的方式，参数为枚举值，Qt.ToolButtonStyle对象：

DockOption对象：1. QMainWindow.AnimatedDocks，设置animated property；2. QMainWindow.AllowNestedDocks，设置是否可以嵌套停靠；3. QMainWindow.AllowTabbedDocks，用户可以将一个dock小部件放在另一个小部件的“上面”。这两个小部件堆叠在一起，并出现一个选项卡栏，用于选择哪一个是可见的。；4. QMainWindow.ForceTabbedDocks，每个dock区域都包含一堆选项卡式dock小部件。换句话说，QDockWidget不能在dock区域中相邻放置。如果设置了此选项，AllowNestedDocks将不起作用；5. QMainWindow.VerticalTabs，主窗口两侧的两个垂直停靠区域垂直显示它们的选项卡。如果未设置此选项，则所有停靠区域都会在底部显示其选项卡；6. QMainWindow.GroupedDragging，当拖动dock的标题栏时，所有与之有选项卡的选项卡都将被拖动；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTabWidget

【继承关系】QtWidgets.QTabWidget -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addTab  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  cornerWidget  count  create  currentChanged  currentIndex  currentWidget  customEvent  destroy  disconnectNotify  documentMode  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  elideMode  enterEvent  event  focusInEvent | focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hasHeightForWidth  heightForWidth  hideEvent  iconSize  indexOf  initPainter  initStyleOption  inputMethodEvent  insertTab  isMovable  isSignalConnected  isTabEnabled  isTabVisible  keyPressEvent  keyReleaseEvent  leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent | moveEvent  nativeEvent  paintEvent  receivers  removeTab  resizeEvent  sender  senderSignalIndex  setCornerWidget  setCurrentIndex  setCurrentWidget  setDocumentMode  setElideMode  setIconSize  setMovable  setTabBar  setTabBarAutoHide  setTabEnabled  setTabIcon  setTabPosition  setTabsClosable  setTabShape  setTabText  setTabToolTip  setTabVisible  setTabWhatsThis | setUsesScrollButtons  sharedPainter  showEvent  sizeHint  tabBar  tabBarAutoHide  tabBarClicked  tabBarDoubleClicked  tabCloseRequested  tabIcon  tabInserted  tabletEvent  tabPosition  tabRemoved  tabsClosable  tabShape  tabText  tabToolTip  tabWhatsThis  timerEvent  updateMicroFocus  usesScrollButtons  wheelEvent  widget |

【详细描述】QTabWidget主要是用来分页显示的，每一页一个界面，众多界面公用一块区域，节省了界面大小，很方便的为用户显示更多的信息

addTab：将QWidget对象添加到tab当中去，参数1. QWidget对象；2.

setTabText：设置页面的名字.

setTabToolTip：设置页面的提示信息，参数为Qstring类型；

setTabEnabled：设置页面是否为被激活状态，参数为bool；

setTabPositio：设置页面名字的位置，参数为枚举值，QTabWidget.TabPosition对象；

setTabsClosable：设置页面关闭按钮，参数为bool；

currentIndex：返回当前页面的下标，从0开始；count：返回页面的数量；

removeTab：删除页面，参数int为页面编号；clear：清空所有页面；

setMoveable：设置页面是否可被拖拽移动，参数为bool；

setCurrentIndex：设置当前显示的页面，参数int为页面编号；

setCurrentWidget：设置当前可见的界面；

insertTab：将一个Tab控件的选项卡插入到指定的位置；

tabCloseRequested：当点击第参数个选项卡的关闭按钮的时候，发出信号，参数为int；

tabBarClicked：当点击第参数个选项卡的时候，发出信号，参数为int；

currentChanged：当改变第参数个选项卡的时候，发出信号，参数为int；

tabBarDoubleClicked：当双击第参数个选项卡的时候，发出信号，参数为int；

TabPosition对象：1. QTabWidget.North；2. QTabWidget.South；3. QTabWidget.West；4. QTabWidget.East；

TabShape对象：1. QTabWidget.Rounded，标签是圆形标签；2. QTabWidget.Triangular，标签是三角形标签；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDockWidget

【继承关系】QtWidgets.QDockWidget -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  allowedAreas  allowedAreasChanged  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dockLocationChanged  dragEnterEvent  dragLeaveEvent  dragMoveEvent | dropEvent  enterEvent  event  features  featuresChanged  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  initStyleOption  inputMethodEvent  isAreaAllowed  isFloating | isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent  sender  senderSignalIndex | setAllowedAreas  setFeatures  setFloating  setTitleBarWidget  setWidget  sharedPainter  showEvent  tabletEvent  timerEvent  titleBarWidget  toggleViewAction  topLevelChanged  updateMicroFocus  visibilityChanged  wheelEvent  widget |

【详细描述】停靠窗口QDockWidget类是一个可以停靠在QMainWindow内的窗口控件，它可以保持在浮动状态或者在指定位置作为子窗口附加到主窗口中，QMainWindow类的主窗口对象保留有一个用于停靠窗口的区域，这个区域在控件的中央周围；

setWidget：在Dock窗口区域设置QWidget；

setFearures：设置停靠窗口的功能属性，参数为枚举值，【DockWidgetClosable，可关闭；DockWidgetMovable，可移动；DockWidgetFloatable，可漂浮；DockWidgetVerticalTitleBar，在左边显示垂直的标签栏；AllDockWidgetFeatures，具有前三种属性的所有功能；NoDockWidgetFeatures，无法关闭，不能悬浮，不能移动】

setFloating：设置Dock窗口是否可以浮动，如果设置为True，则表示可以浮动；

setAlllowedAreas：设置窗口可以停靠的区域，参数为枚举值，【LeftDockWidgetArea，左侧停靠区域；RightDockWidgetArea，右侧停靠区域；TopDockWidgetArea，顶部停靠区域；BottomDockWidgetArea，底部停靠区域；NoDockWidgetArea，不显示Widget】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMenu

【继承关系】QtWidgets.QMenu -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| aboutToHide  aboutToShow  actionAt  actionEvent  actionGeometry  activeAction  addAction  addMenu  addSection  addSeparator  changeEvent  childEvent  clear  closeEvent  columnCount  connectNotify  contextMenuEvent  create  customEvent  defaultAction  destroy  disconnectNotify | dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  exec  exec\_  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  hideTearOffMenu  hovered  icon  initPainter  initStyleOption  inputMethodEvent  insertMenu  insertSection | insertSeparator  isEmpty  isSignalConnected  isTearOffEnabled  isTearOffMenuVisible  keyPressEvent  keyReleaseEvent  leaveEvent  menuAction  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  popup  receivers  resizeEvent  sender  senderSignalIndex | separatorsCollapsible  setActiveAction  setDefaultAction  setIcon  setNoReplayFor  setSeparatorsCollapsible  setTearOffEnabled  setTitle  setToolTipsVisible  sharedPainter  showEvent  showTearOffMenu  sizeHint  tabletEvent  timerEvent  title  toolTipsVisible  triggered  updateMicroFocus  wheelEvent |

【详细描述】QMenu类提供了一个可以添加到菜单栏的小部件，也用于创建上下文菜单和弹出菜单。每个QMenu对象可以包含一个或多个QAction对象或级联的QMenu对象。

exec\_：参数为pos对象

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMenuBar

【继承关系】QtWidgets.QMenuBar -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionAt  actionEvent  actionGeometry  activeAction  addAction  addMenu  addSeparator  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  cornerWidget  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent | dragMoveEvent  dropEvent  enterEvent  event  eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  heightForWidth  hideEvent  hovered  initPainter  initStyleOption  inputMethodEvent  insertMenu  insertSeparator  isDefaultUp  isNativeMenuBar | isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent  sender  senderSignalIndex  setActiveAction  setCornerWidget  setDefaultUp | setNativeMenuBar  setSeparatorsCollapsible  setTearOffEnabled  setTitle  setToolTipsVisible  setVisible  sharedPainter  showEvent  showTearOffMenu  sizeHint  tabletEvent  timerEvent  title  toolTipsVisible  triggered  updateMicroFocus  wheelEvent |

【详细描述】QMenuBar菜单栏，位于 QMainWindow 对象的标题栏下方。提供了一个可以添加到菜单栏的小控件，每个 QMenu 对象都可以包含一个或多个 QAction 对象或级联的 QMenu 对象。QMenu 也可以用来创建上下文菜单和弹出菜单。

addMenu：在菜单栏中添加一个新的QMenu对象，返回QMenu对象；

addAction：向QMenu小控件中添加一个操作按钮，其中包含文本或图标；

setEnabled：将操作按钮设置为启用/禁用

addSeperator：在菜单中添加一条分割线

clear：删除菜单栏的内容

setText：设置菜单项的文本；text：返回与QAction对象关联的文本

setTitle：设置QMenu小控件的标题

title：返回QMenu小控件的标题

triggered：单击任何QAction按钮时，QMenu对象会发射一个triggered信号，该函数会接收信号的QAction对象；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStatusBar

【继承关系】QtWidgets.QStatusBar -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addPermanentWidget  addWidget  changeEvent  childEvent  clearMessage  closeEvent  connectNotify  contextMenuEvent  create  currentMessage  customEvent  destroy  disconnectNotify  dragEnterEvent | dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  hideOrShow  initPainter  inputMethodEvent  insertPermanentWidget | insertWidget  isSignalConnected  isSizeGripEnabled  keyPressEvent  keyReleaseEvent  leaveEvent  messageChanged  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent | receivers  reformat  removeWidget  resizeEvent  sender  senderSignalIndex  setSizeGripEnabled  sharedPainter  showEvent  showMessage  tabletEvent  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】MainWindow对象在底部保留一个水平条，作为QStatusBar，状态栏是用来显示应用的状态信息的组件；

addWidget：在状态栏中添加给定的窗口小控件对象；

addPermanentWidget：在状态栏中永久添加给定的窗口小控件对象；

showMessage：在状态栏上显示一条信息；

clearMessage：删除正在显示的临时信息；

removeWidget：从状态栏中移除指定的小控件；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QToolBar

【继承关系】QtWidgets.QToolBar -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionAt  actionEvent  actionGeometry  actionTriggered  addAction  addSeparator  addWidget  allowedAreas  allowedAreasChanged  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent | dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  iconSize  iconSizeChanged  initPainter  initStyleOption  inputMethodEvent  insertSeparator  insertWidget  isAreaAllowed  isFloatable | isFloating  isMovable  isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  movableChanged  moveEvent  nativeEvent  orientation  orientationChanged  paintEvent  receivers  resizeEvent  sender | senderSignalIndex  setAllowedAreas  setFloatable  setIconSize  setMovable  setOrientation  setToolButtonStyle  sharedPainter  showEvent  tabletEvent  timerEvent  toggleViewAction  toolButtonStyle  toolButtonStyleChanged  topLevelChanged  updateMicroFocus  visibilityChanged  wheelEvent  widgetForAction |

【详细描述】QToolBar是由文本按钮，图标或其他小空间按钮组成的可移动面板，通常位于菜单栏下方；

addAction：添加具有文本或图标的工具按钮；

addSeperator：分组显示工具按钮

addWidget：添加工具栏中按钮以外的控件

setMovable：工具变得可移动

setFloatable： 工具条可浮动(True:可浮动， False,不可浮动)；

setOrientation：工具栏的方向可以设置为Qt.Horizontal或Qt.Vertical；

setAllowedAreas：设置工具条允许停靠的区域。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QProgressBar

【继承关系】QtWidgets.QProgressBar -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  alignment  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  focusInEvent | focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  format  hideEvent  initPainter  initStyleOption  inputMethodEvent  isSignalConnected  isTextVisible  keyPressEvent  keyReleaseEvent  leaveEvent  maximum  metric  minimum  minimumSizeHint | mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  orientation  paintEvent  receivers  reset  resetFormat  resizeEvent  sender  senderSignalIndex  setAlignment  setFormat  setInvertedAppearance  setMaximum | setMinimum  setOrientation  setRange  setTextDirection  setTextVisible  setValue  sharedPainter  showEvent  sizeHint  tabletEvent  text  timerEvent  updateMicroFocus  value  valueChanged  wheelEvent |

【详细描述】QProgressBar，进度条控件，提供一个水平或垂直进度条，用于向用户提供操作进度的提示，并向他们保证相应应用程序仍在运行。

setRange：设置进度条范围；setMinimum：设置最小值；setMaximum：设置最大值；

setValue：设置当前值；

## QAction

【继承关系】QtWidget.QAction -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| actionGroup  activate  associatedGraphicsWidgets  associatedWidgets  autoRepeat  changed  childEvent  connectNotify  customEvent  data  disconnectNotify  event  font  hover  hovered  icon  iconText  isCheckable | isChecked  isEnabled  isIconVisibleInMenu  isSeparator  isShortcutVisibleInContextMenu  isSignalConnected  isVisible  menu  menuRole  parentWidget  priority  receivers  sender  senderSignalIndex  setActionGroup  setAutoRepeat  setCheckable  setChecked | setData  setDisabled  setEnabled  setFont  setIcon  setIconText  setIconVisibleInMenu  setMenu  setMenuRole  setPriority  setSeparator  setShortcut  setShortcutContext  setShortcuts  setShortcutVisibleInContextMenu  setStatusTip  setText  setToolTip | setVisible  setWhatsThis  shortcut  shortcutContext  shortcuts  showStatusText  statusTip  text  timerEvent  toggle  toggled  toolTip  trigger  triggered  whatsThis |

【详细描述】QAction类提供了抽象的用户界面action，这些action可以被放置在窗口部件中。应用程序可以通过菜单，工具栏按钮以及键盘快捷键来调用通用的命令。QActions可以被添加到菜单和工具栏中，并且可以在菜单和工具栏中同步更新。例如，在一个字处理软件中，如果用户在工具栏中按下了Bold按钮，那么菜单中的Bold选项就会自动被选中。QActions可以作为独立的对象被创建，但是我们也可以在构建菜单的时候创建它们；QMenu类包含了非常简便的方法用于创建适合用作菜单项的QActions。

可以通过QWidget.addAction()或者是QGraphicsWidget.addAction()函数将Actions添加到窗口部件上。注意，只有将QActions添加到窗口部件上之后，我们才可以使用这些actions；当actions的快捷键是全局的时候，我们也必须先将actions添加到窗口部件上。

一旦QAction被创建了，那么就必须将它添加到相关的菜单和工具栏上，然后将它们链接到实现相应action功能的槽函数上。

初始化：a = QAction（QIcon, str，self）, str ：动作关联的部件对象，QAction中使用&可以设置快捷键，格式：WidgetName(&X)

setIcon：设置图标；

setToolTip：设置提示字体；

setStatusTip：设置状态提示；

setShortcut：将快捷键关联到操作按钮；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QWidgetAction

【继承关系】QtWidget.QWidgetAction -> QtWidget.QAction -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  connectNotify  createdWidgets  createWidget  customEvent | defaultWidget  deleteWidget  disconnectNotify  event  eventFilter | isSignalConnected  receivers  releaseWidget  requestWidget  sender | senderSignalIndex  setDefaultWidget  timerEvent |

【详细描述】通过一个接口扩展QAction，将定制的小部件插入到基于操作的容器中；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QActionGroup

【继承关系】QtWidget. QActionGroup -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| actions  addAction  checkedAction  childEvent  connectNotify  customEvent | disconnectNotify  exclusionPolicy  hovered  isEnabled  isExclusive  isSignalConnected | isVisible  receivers  removeAction  sender  senderSignalIndex  setDisabled | setEnabled  setExclusionPolicy  setExclusive  setVisible  timerEvent  triggered |

【详细描述】将操作分组在一起；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDesktopWidget

【继承关系】QtWidgets.QDesktopWidget -> QtWigets.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  availableGeometry  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent | enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  inputMethodEvent  isSignalConnected  isVirtualDesktop  keyPressEvent  keyReleaseEvent  leaveEvent | metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  primaryScreen  primaryScreenChanged  receivers  resized  resizeEvent  screen  screenCount | screenCountChanged  screenGeometry  screenNumber  sender  senderSignalIndex  sharedPainter  showEvent  tabletEvent  timerEvent  updateMicroFocus  wheelEvent  workAreaResized |

【详细描述】用于描述屏幕的类，提供有关屏幕的信息，例如屏幕大小，数量，以及它们是配置为单独的桌面还是单个虚拟桌面；此外，由Qt提供的小工具使用这个类来将工具提示，菜单和对话框放置在他们的父应用程序或应用程序小部件的正确屏幕上；应用程序可以使用此类来获取可用于保存窗口位置的信息，或将子窗口小部件和对话框放置在一个特定屏幕上。

availableGeometry：获取可用位置和尺寸（除去任务栏等），返回屏幕的QRect对象，参数为显示屏索引，如果安装了两个显示屏，主显示屏索引为0，辅显示屏索引为1；

screenGeometry：获取屏幕的分辨率，参数为显示屏索引，如果安装了两个显示屏，主显示屏索引为0，辅显示屏索引为1；

screenCount：获取显示器数量；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QLabel

【继承关系】QtWidgets.QLabel -> QtWidgets.Frame -> QtWidgets. QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  alignment  buddy  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  enterEvent  event  focusInEvent  focusNextChild | focusNextPrevChild  focusOutEvent  focusPreviousChild  hasScaledContents  hasSelectedText  heightForWidth  hideEvent  indent  initPainter  initStyleOption  inputMethodEvent  isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  linkActivated  linkHovered  margin  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent | mousePressEvent  mouseReleaseEvent  moveEvent  movie  nativeEvent  openExternalLinks  paintEvent  picture  pixmap  receivers  resizeEvent  selectedText  selectionStart  sender  senderSignalIndex  setAlignment  setBuddy  setIndent  setMargin  setMovie  setNum  setOpenExternalLinks | setPicture  setPixmap  setScaledContents  setSelection  setText  setTextFormat  setTextInteractionFlags  setWordWrap  sharedPainter  showEvent  sizeHint  tabletEvent  text  textFormat  textInteractionFlags  timerEvent  updateMicroFocus  wheelEvent  wordWrap |

【详细描述】可用于显示不可编辑的文本或图片，或GIF动画，还可用于提示标记为其他控件。

setAlignment：按固定方式对齐文本，参数为枚举值，参见Qt.AlignmentFlag对象

setIndent：设置文本缩进值；

setPixmap：设置QLabel上显示一个Pixmap图片，参数为QPixmap对象；

setMovie：设置在QLabel上显示一个动图（.gif文件），参数为路径；

setScaledContents：设置 label 自动缩放其内容，以填充所有可用空间，参数为bool；hasScaledContents：查看是否设置了自动缩放内容以填充所有可用空间；

text：获取QLabel的文本内容；setText：设置文本内容；

selectedText：返回所选择的字符；

setBuddy：设置伙伴关系，参数为伙伴对象；

setWordWrap：设置是否允许换行；

setMargin：设置 label 外边距，即文本内容到边框的距离（以像素为单位）；

setTextInteractionFlags：设置 label 在显示文本时，应如何与用户输入交互，参数为枚举值，参见Qt.TextInteractionFlag对象；

setOpenExternalLinks：设置打开的超链接是否在新窗口中打开；

linkActivated：单击标签，打开超链接【信号】；

linkHovered：当鼠标划过标签所需做的动作【信号】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QLineEdit

【继承关系】QtWidget.QLineEdit -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| addAction  alignment  backspace  changeEvent  clear  completer  contextMenuEvent  copy  createStandardContextMenu  cursorBackward  cursorForward  cursorMoveStyle  cursorPosition  cursorPositionAt  cursorPositionChanged  cursorRect  cursorWordBackward  cursorWordForward  cut  del\_  deselect  displayText  dragEnabled | dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  echoMode  editingFinished  end  event  focusInEvent  focusOutEvent  getTextMargins  hasAcceptableInput  hasFrame  hasSelectedText  home  initStyleOption  inputMask  inputMethodEvent  inputMethodQuery  inputRejected  insert  isClearButtonEnabled  isModified | isReadOnly  isRedoAvailable  isUndoAvailable  keyPressEvent  maxLength  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  paintEvent  paste  placeholderText  redo  returnPressed  selectAll  selectedText  selectionChanged  selectionEnd  selectionLength  selectionStart  setAlignment  setClearButtonEnabled | setCompleter  setCursorMoveStyle  setCursorPosition  setDragEnabled  setEchoMode  setFrame  setInputMask  setMaxLength  setModified  setPlaceholderText  setReadOnly  setSelection  setText  setTextMargins  setValidator  sizeHint  text  textChanged  textEdited  textMargins  undo  validator |

【详细描述】用于输入输出单行字符的组件，也可用于输入输出数值数据（需要进行字符串与数字之间的转换），所涉及到多行，则使用QTextEdit；

【常用函数】

setAlignment：按固定值方式对齐文本，参数为枚举值，参见Qt.AlignmentFlag对象

clear：清除文本框内容；

SetPlaceholderText：设置文本框浮显的文字内容

placeholderText：返回浮显的文本内容

setMaxLength：设置文本框所允许输入的最大字符数；

setReadOnly：设置文本框为只读的；

SetText：设置文本框内容；

text：返回文本框内容；

setDragEnabled：设置QWidget对象A可以拖动，参数为Bool；dragEnabled：返回当前QLineEdit对象是否可以被拖拽；

setMaxLength：设置允许输入字符的最大长度；

selectAll：全选；

setFocus：获得焦点

setEchoMode：设置文本回显，用于设置允许输入的文本显示格式的参数【QLineEdit.Normal，正常显示所输入的字符；QLineEdit.NoEcho，不显示任何输入的字符，常用于密码类型的输入，且其密码长度需要保密时；QLineEdit.Password，显示与平台相关的密码掩码字符，而不是实际输入的字符；QLineEdit.PasswordEchoOnEdit，在编辑时显示字符，负责显示密码类型的输入】

SetValidator：设置文本框的验证器（验证规则），将限制任意可能输入的文本，参数为【QIntValidator对象，限制输入整数；QDoubleValidator对象，限制输入浮点数；QRegexpValidator对象，检查输入是否符合正则表达式】

setInputMask：设置掩码，参数为掩码占位符；

selectionChanged：选择改变触发信号；

textChanged：文本内容修改，触发信号；

editingFinished：文本结束编辑，触发信号；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTextEdit

【继承关系】QtWidget.QTextEdit -> QtWidget.QAbstactScrollArea -> QtWidget.QFrame -> QtWidget.QWidget

【继承者】QTextBrowser

|  |  |  |  |
| --- | --- | --- | --- |
| acceptRichText  actionEvent  alignment  anchorAt  append  autoFormatting  canInsertFromMimeData  canPaste  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  copy  copyAvailable  create  createMimeDataFromSelection  createStandardContextMenu  currentCharFormat  currentCharFormatChanged  currentFont  cursorForPosition  cursorPositionChanged  cursorRect  cursorWidth  customEvent  cut  destroy  disconnectNotify  document  documentTitle  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  ensureCursorVisible  enterEvent | event  eventFilter  extraSelections  find  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  fontFamily  fontItalic  fontPointSize  fontUnderline  fontWeight  hideEvent  initPainter  initStyleOption  inputMethodEvent  inputMethodQuery  insertFromMimeData  insertHtml  insertPlainText  isReadOnly  isSignalConnected  isUndoRedoEnabled  keyPressEvent  keyReleaseEvent  leaveEvent  lineWrapColumnOrWidth  lineWrapMode  loadResource  mergeCurrentCharFormat  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent | nativeEvent  overwriteMode  paintEvent  paste  placeholderText  print  print\_  receivers  redo  redoAvailable  resizeEvent  scrollContentsBy  scrollToAnchor  selectAll  selectionChanged  sender  senderSignalIndex  setAcceptRichText  setAlignment  setAutoFormatting  setCurrentCharFormat  setCurrentFont  setCursorWidth  setDocument  setDocumentTitle  setExtraSelections  setFontFamily  setFontItalic  setFontPointSize  setFontUnderline  setFontWeight  setHtml  setLineWrapColumnOrWidth  setLineWrapMode  setMarkdown  setOverwriteMode  setPlaceholderText  setPlainText  setReadOnly | setTabChangesFocus  setTabStopDistance  setTabStopWidth  setText  setTextBackgroundColor  setTextColor  setTextCursor  setTextInteractionFlags  setUndoRedoEnabled  setViewportMargins  setWordWrapMode  sharedPainter  showEvent  tabChangesFocus  tabletEvent  tabStopDistance  tabStopWidth  textBackgroundColor  textChanged  textColor  textCursor  textInteractionFlags  timerEvent  toHtml  toMarkdown  toPlainText  undo  undoAvailable  updateMicroFocus  viewportEvent  viewportMargins  viewportSizeHint  wheelEvent  wordWrapMode  zoomIn  zoomOut |

【详细描述】多行文本输入输出，其输入的参数为QTextDocument对象

【常用函数】

acceptRichText：此属性用来指定文本框是否接受富文本输入。当设置为false时，文本框只接受纯文本；

setAlignment：按固定值方式对齐文本，参数为枚举值，参见Qt.AlignmentFlag对象；

setFontFamily：设置字体；fontFamily：获取字体（可以用QFontDialog.getFont()来查看可用的字体和字体能使用的效果）

setFontWeight：设置字体粗细，参数QFont.Bold，或大小；fontWeight：返回字体粗细大小；

setFontPointSize：设置字体的尺寸（float）；fontPointSize：获取字体的尺寸，返回float；

setFontItalic：设置斜体，参数True/False；fontItalic：获取字体设定是否为斜体的判断，返回Bool；

setFontUnderline：设置下划线，参数为Bool；fontUnderline：获取是否启用下划线的判断，返回Bool；

setCurrentFont：对当前字体进行统一设置，参数为QFont对象；

setTextBackgroundColor：设置字体的背景色，参数为QColor对象；textBackgroundColor：获取字体的背景色，返回QColor对象；

setTextColor：设置字体的颜色，参数为QColor对象；textColor：获取字体的颜色，返回QColor对象；

setCurrentCharFormat：通过QTextCharFormat对字体进行设置，参数为QTextCharFormat对象；

copy：复制

paste：粘贴

canPaste：返回是否可以从剪切板中粘贴文本到编辑器中，返回bool对象；

setUndoRedoEnabled：此属性用来指定撤销（ctrl+z）和恢复（ctrl+y）功能是否可用，参数为bool类型；

undo/redo：可以撤销或者恢复上一次操作。

selectAll：可以选择（或者取消选择）整个文本；

setTextInteractionFlags：指定控件如何响应用户输入，参数为枚举值，参见Qt.TextInteractionFlag对象；textInteractionFlags：返回内容

setReadOnly：设置位只读模式；isReadOnly：返回判断；

setTabChangesFocus：设置tab键是切换焦点，还是作为输入。默认为false。设置为true后，若存在其他控件，则按tab键后，焦点将会切换到另一个控件，若不存在其他控件，tab还是作为输入。tabChangesFocus，返回是否已tab键作为切换焦点；

setTabStopDistance：此属性用来指定tab键对应的像素值，默认为80px。tabStopDistance：返回判断；

find：查找某个文本，参数为str, options =【QTextDocument.FindBackward（从后向前查找），QTextDocument.FindCaseSensitively（默认查找不区分大小写，使用该枚举值启用区分大小写查找），QTextDocument.FindWholeWords（找完整的单词）】

setPlainText：将普通文本内容显示到控件中；toPlainText：获取对应控件的普通文本；insertPlainText：在光标处插入普通文本，第一个参数为内容，第二个参数为文本光标；

setHtml：按html文本格式显示到控件中；toHtml：获取对应控件的html文本；insertHtml：在光标处插入html文本；

SetPlaceholderText：设置文本框浮显的文字内容；

placeholderText：返回浮显的文本内容；

append：追加文本；

clear：文本清除；

setCursorWidth：设定光标宽度；cursorWidth：获取光标宽度，返回int；

TextCursor：获取文本光标；’

cursorRect：获取光标区间，返回QRect对象；

setAutoFormatting：设置自动格式化【QTextEdit.AutoNone 不进行任何自动格式化；QTextEdit.AutoBulletList 自动创建列表，(当用户在当前行最左侧输入一个星号\*，就会创建一个列表条目。如果在一个已有的列表后输入回车，则会创建一个新列表，这个功能类似于markdown的语法）；QTextEdit.AutoAll 打开所有自动格式化，但目前只支持列表的自动格式化】

setLineWrapMode：设置换行模式【QTextEdit.NoWrap 不换行；QTextEdit.FixedPixelWidth 按像素限制换行；FixedColumnWidth 按列数限制换行（若使用按像素或列数限制，则需要通过setLineWrapColumnOrWidth定义换行的像素宽度或字符数宽度）】；lineWrapMode：返回换行模式

setWordWrapMode：设置单词换行模式【QTextOption.WordWrap 可保持单词的完整性，其他参加QTextOption参数】；wordWrapMode：获取单词换行模式；

setOverwriteMode：设置覆盖模式，参数为True/False；overwriteMode：返回是否位覆盖模式的判断，返回bool；

setDocumentTitle：保存从文档文本解析出的文档题目。

#setText：根据参数内容自动识别是纯文本也可以是html文本；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPlainTextEdit

【继承关系】QtWidget.QPlainTextEdit -> QtWidget.QAbstactScrollArea -> QtWidget.QFrame -> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  anchorAt  appendHtml  appendPlainText  backgroundVisible  blockBoundingGeometry  blockBoundingRect  blockCount  blockCountChanged  canInsertFromMimeData  canPaste  centerCursor  centerOnScroll  changeEvent  childEvent  clear  closeEvent  connectNotify  contentOffset  contextMenuEvent  copy  copyAvailable  create  createMimeDataFromSelection  createStandardContextMenu  currentCharFormat  cursorForPosition  cursorPositionChanged  cursorRect  cursorWidth  customEvent  cut  destroy  disconnectNotify  document | documentTitle  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  ensureCursorVisible  enterEvent  event  eventFilter  extraSelections  find  firstVisibleBlock  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  getPaintContext  hideEvent  initPainter  initStyleOption  inputMethodEvent  inputMethodQuery  insertFromMimeData  insertPlainText  isReadOnly  isSignalConnected  isUndoRedoEnabled  keyPressEvent  keyReleaseEvent  leaveEvent  lineWrapMode  loadResource  maximumBlockCount | mergeCurrentCharFormat  metric  modificationChanged  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  overwriteMode  paintEvent  paste  placeholderText  print  print\_  receivers  redo  redoAvailable  resizeEvent  scrollContentsBy  selectAll  selectionChanged  sender  senderSignalIndex  setBackgroundVisible  setCenterOnScroll  setCurrentCharFormat  setCursorWidth  setDocument  setDocumentTitle  setExtraSelections  setLineWrapMode  setMaximumBlockCount  setOverwriteMode | setPlaceholderText  setPlainText  setReadOnly  setTabChangesFocus  setTabStopDistance  setTabStopWidth  setTextCursor  setTextInteractionFlags  setUndoRedoEnabled  setViewportMargins  setWordWrapMode  sharedPainter  showEvent  tabChangesFocus  tabletEvent  tabStopDistance  tabStopWidth  textChanged  textCursor  textInteractionFlags  timerEvent  toPlainText  undo  undoAvailable  updateMicroFocus  updateRequest  viewportEvent  viewportMargins  viewportSizeHint  wheelEvent  wordWrapMode  zoomIn  zoomOut |

【详细描述】QPlainTextEdit 是一个多行文本编辑器，用于显示和编辑多行简单文本。由于QPlainTextEdit是用逐行逐段滚动的方法替代了像素精准度高的滚动方法，所以效率比QTextEdit要更高一些（每次滚动都需要重新的绘制），在处理大批量的文本内容时，比QTextEdit有更强大的优势。

toPlainText：将全部文字内容输出为一个字符串；

document：返回QTextDocument对象的指针；

blockCount：获得文本块个数

setPlaceholderText：提示占位文本

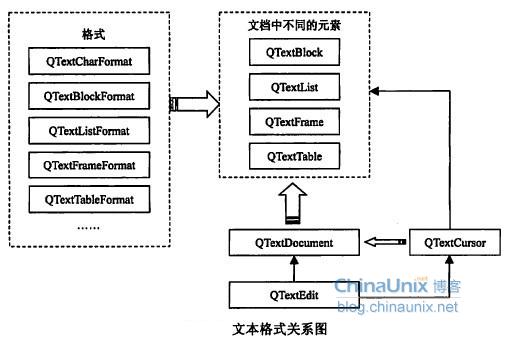
#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTextDocument

【继承关系】QtGui.QTextDocument -> QtGui.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| addResource  adjustSize  allFormats  availableRedoSteps  availableUndoSteps  baseUrl  baseUrlChanged  begin  blockCount  blockCountChanged  characterAt  characterCount  childEvent  clear  clearUndoRedoStacks  clone  connectNotify  contentsChange  contentsChanged  createObject  cursorPositionChanged  customEvent  defaultCursorMoveStyle  defaultFont | defaultStyleSheet  defaultTextOption  disconnectNotify  documentLayout  documentLayoutChanged  documentMargin  drawContents  end  find  findBlock  findBlockByLineNumber  findBlockByNumber  firstBlock  idealWidth  indentWidth  isEmpty  isModified  isRedoAvailable  isSignalConnected  isUndoAvailable  isUndoRedoEnabled  lastBlock  lineCount  loadResource | markContentsDirty  maximumBlockCount  metaInformation  modificationChanged  object  objectForFormat  pageCount  pageSize  print  print\_  receivers  redo  redoAvailable  resource  revision  rootFrame  sender  senderSignalIndex  setBaseUrl  setDefaultCursorMoveStyle  setDefaultFont  setDefaultStyleSheet  setDefaultTextOption  setDocumentLayout | setDocumentMargin  setHtml  setIndentWidth  setMarkdown  setMaximumBlockCount  setMetaInformation  setModified  setPageSize  setPlainText  setTextWidth  setUndoRedoEnabled  setUseDesignMetrics  size  textWidth  timerEvent  toHtml  toMarkdown  toPlainText  toRawText  undo  undoAvailable  undoCommandAdded  useDesignMetrics |

【详细描述】富文本格式，简单来说就是在文档中可以使用多种格式，比如字体颜色、图片和表格等等。它是与纯文本（Plain Text）相对而言的，比如Windows上的记事本就是纯文本编辑器，而Word就是富文本编辑器。富文本的处理分为了编辑操作和只读操作两种方式。编辑操作是使用基于光标（QTextCursor）的一些接口函数， 而对于文档结构的概览，使用了只读的分层次的接口函数（QTextDocument），它们有利于文档的检索和输出。富文本文档的结构被分为了几种元素来表示【QTextFrame（框架），QTextBlock（文本块）、QTextTable（表格），QTextList（列表）】，而每种元素的格式又使用相应的format类来表示【QTextFrameFormat（框架格式），QTextBlockFormat（文本块格式），QTextTableFormat（表格格式），QTextListFormat（列表格式）】，而文本块的格式由QTextBlockFormat类来处理，它主要涉及对齐方式，文本块四周的边白，缩进等内容。而文本块中的文本内容的格式，比如字体大小、加粗、下划线等内容，则由QTextCharFormat类来设置。

[](http://blog.chinaunix.net/attachment/201209/12/25806493_1347445833iK9u.jpg)

从上图可知，QTextCursor类是一个非常重要也经常会用到的类，它用于表示编辑文本中的光标。这个类提供了对QTextDocument文档的修改接口，所有对文档格式的修改，说到底都与光标有关，如改变字符的格式，指的是改变光标处字符的格式；改变段落的格式，指的是改变光标所在段落的格式，因此，QTextCursor类在文档编辑类程序中有着重要的作用，所有对QTextDocument的修改能够通过QTextCursor类实现。

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

## QTextCursor

|  |  |  |  |
| --- | --- | --- | --- |
| anchor  atBlockEnd  atBlockStart  atEnd  atStart  beginEditBlock  block  blockCharFormat  blockFormat  blockNumber  charFormat  clearSelection  columnNumber  createList  currentFrame | currentList  currentTable  deleteChar  deletePreviousChar  document  endEditBlock  hasComplexSelection  hasSelection  insertBlock  insertFragment  insertFrame  insertHtml  insertImage  insertList  insertTable | insertText  isCopyOf  isNull  joinPreviousEditBlock  keepPositionOnInsert  mergeBlockCharFormat  mergeBlockFormat  mergeCharFormat  movePosition  position  positionInBlock  removeSelectedText  select  selectedTableCells  selectedText | selection  selectionEnd  selectionStart  setBlockCharFormat  setBlockFormat  setCharFormat  setKeepPositionOnInsert  setPosition  setVerticalMovementX  setVisualNavigation  swap  verticalMovementX  visualNavigation |

【详细描述】QTextCursor是专门用来协助处理和获取QTextDocument内容的api. 它包含的信息既有QTextDocument内的光标位置（position()），以及所做的selection（anchor() 和 position()之间）。current character是指position之前的那个character. current block是指包含position()位置的block.

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTextCharFormat

【继承关系】QtGui. QTextCharFormat -> QtGui. QTextFormat

|  |  |  |  |
| --- | --- | --- | --- |
| anchorHref  anchorNames  font  fontCapitalization  fontFamilies  fontFamily  fontFixedPitch  fontHintingPreference  fontItalic  fontKerning  fontLetterSpacing  fontLetterSpacingType  fontOverline  fontPointSize  fontStretch  fontStrikeOut | fontStyleHint  fontStyleName  fontStyleStrategy  fontUnderline  fontWeight  fontWordSpacing  isAnchor  isValid  setAnchor  setAnchorHref  setAnchorNames  setFont  setFontCapitalization  setFontFamilies  setFontFamily  setFontFixedPitch | setFontHintingPreference  setFontItalic  setFontKerning  setFontLetterSpacing  setFontLetterSpacingType  setFontOverline  setFontPointSize  setFontStretch  setFontStrikeOut  setFontStyleHint  setFontStyleName  setFontStyleStrategy  setFontUnderline  setFontWeight  setFontWordSpacing  setTableCellColumnSpan | setTableCellRowSpan  setTextOutline  setToolTip  setUnderlineColor  setUnderlineStyle  setVerticalAlignment  tableCellColumnSpan  tableCellRowSpan  textOutline  toolTip  underlineColor  underlineStyle  verticalAlignment |

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractButton

【继承关系】QtWidgets.QAbstractButton -> QtWidgets.QWidget

【继承者】QCheckBox，QPushButton，QRadioButton，QToolButton，QCommandLinkButton

|  |  |  |  |
| --- | --- | --- | --- |
| animateClick（Slot）  autoExclusive（F）  autoRepeat  autoRepeatDelay  autoRepeatInterval  changeEvent  checkStateSet（VF）  click  clicked[Sig]  event  focusInEvent | focusOutEvent  group  hitButton（VF）  icon  iconSize  isCheckable  isChecked  isDown  keyPressEvent  keyReleaseEvent  mouseMoveEvent | mousePressEvent  mouseReleaseEvent  nextCheckState（VF）  paintEvent  pressed[Sig]  released[Sig]  setAutoExclusive  setAutoRepeat  setAutoRepeatDelay  setAutoRepeatInterval  setCheckable | setChecked  setDown  setIcon  setIconSize  setShortcut  setText  shortcut  text  timerEvent  toggle  toggled[Sig] |

【详细描述】

setCheckable：设置按钮是否已经被选中，如果设置为True，则表示按钮将保持已点击状态；

toggle：在按钮状态之间进行切换；

setIcon：设置按钮上的图表；

setEnabled：设置按钮是否可以使用，当设置为False时，按钮变成不可用状态，点击它不会发射信号（继承自QWidget）；

isDown：提示按钮是否被按下；

isChecked：返回按钮的状态，返回值为True或者False；

isEnabled：提示按钮是否可以被用户点击（继承自QWidget）；

isCheckAble：提示按钮是否为标记的；

setAutoRepeat：设置按钮是否在用户长按时可以自动重复执行；

setText：设置按钮的显示文本；

text：返回按钮的显示文本；

Pressed：当鼠标指针在按钮上并按下左键时触发该信号；

Released：当鼠标左键被释放是触发该信号；

Clicked：当鼠标左键被按下然后释放时，或者快捷键被释放时触发该信号；

Toggled：当按钮的标记状态发生改变时触发该信号；

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

## QPushButton

【继承关系】QtWidget.QPushButton -> QtWidget.QAbstractButton -> QtWidget. QWidget

【继承者】QCommandLinkButton

|  |  |  |  |
| --- | --- | --- | --- |
| autoDefault  event  focusInEvent  focusOutEvent  hitButton | initStyleOption  isDefault  isFlat  keyPressEvent  menu | minimumSizeHint  paintEvent  setAutoDefault  setDefault  setFlat | setMenu  showMenu  sizeHint |

【详细描述】(按钮)

setDefault：设置按钮的默认状态，外框为蓝色；autoDefault：返回是否开启Default的判断；

-----------------------------------------------------------------------------------------------------------------------------------------------------------------

## QRadioButton

【继承关系】QtWidget.QPushButton -> QtWidget.QAbstractButton -> QtWidget. QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| event  hitButton | initStyleOption[F]  minimumSizeHint | mouseMoveEvent  paintEvent | sizeHint |

【详细描述】（单选框）

## QCheckBox

【继承关系】QtWidget.QCheckBox -> QtWidget.QAbstractButton -> QtWidget. QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| checkState[F]  checkStateSet  event  hitButton | initStyleOption[F]  isTristate[F]  minimumSizeHint  mouseMoveEvent | nextCheckState  paintEvent  setCheckState[Fun]  setTristate[Fun] | sizeHint  stateChanged[Si] |

【详细描述】（复选框）

setTristate：设置复选框可用于半选中状态【True或False】；

setCheckState：设置当前复选框状态，参数为枚举值，参见Qt.CheckState对象

stateChanged：当状态发生变化时触发该信号；

## QComboBox

【继承关系】QtWidget. QComboBox -> QtWidget. QWidget

【继承者】QFontComboBox

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  activated  addItem  addItems  changeEvent  childEvent  clear  clearEditText  closeEvent  completer  connectNotify  contextMenuEvent  count  create  currentData  currentIndex  currentIndexChanged  currentText  currentTextChanged  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  duplicatesEnabled  editTextChanged  enterEvent  event | findData  findText  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hasFrame  hideEvent  hidePopup  highlighted  iconSize  initPainter  initStyleOption  inputMethodEvent  inputMethodQuery  insertItem  insertItems  insertPolicy  insertSeparator  isEditable  isSignalConnected  itemData  itemDelegate  itemIcon  itemText  keyPressEvent  keyReleaseEvent  leaveEvent  lineEdit | maxCount  maxVisibleItems  metric  minimumContentsLength  minimumSizeHint  model  modelColumn  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  placeholderText  receivers  removeItem  resizeEvent  rootModelIndex  sender  senderSignalIndex  setCompleter  setCurrentIndex  setCurrentText  setDuplicatesEnabled  setEditable  setEditText  setFrame  setIconSize  setInsertPolicy | setItemData  setItemDelegate  setItemIcon  setItemText  setLineEdit  setMaxCount  setMaxVisibleItems  setMinimumContentsLength  setModel  setModelColumn  setPlaceholderText  setRootModelIndex  setSizeAdjustPolicy  setValidator  setView  sharedPainter  showEvent  showPopup  sizeAdjustPolicy  sizeHint  tabletEvent  textActivated  textHighlighted  timerEvent  updateMicroFocus  validator  view  wheelEvent |

【详细描述】下拉列表框，是一个集按钮和下拉选项与一体的控件；

addItem：用于添加一个列表项，支持QIcon参数，生成带图标的下拉选项；如果只是添加字符串列表项，而且数据来源于一个 QStringList 变量，可以使用 addltems函数；

itemIcon，itemText，itemData：返回指定索引处项目的图标对象，文本，userdata；

currentIndex，：获取当前项目的索引值；currentText：获取当前项目的文本；currentData：返回当前项的关联数据，参数role为数据角色，其默认值为Qt.UserRole，可以为一个项定义多个角色的用户数据，更多自定义角色的编号从Qt.UserRole开始增加，如定义Qt.UserRole+1，Qt.UserRole+2；

findText：返回寻找内容的索引；

count：返回当前列表框中选项数量；

clear：清空组合框项

clearEditText：清除当前显示内容；

showPopup：弹出项目列表；

setInsertPolicy：设置插入的策略，参数为InsertAfterCurrent，InsertAlphabetically，InsertAtBottom，InsertAtCurrent，InsertAtTop，InsertBeforeCurrent，NoInsert；

insertItem，insertItems：在第i个位置插入条目，参数为i（填充组合框）；

removeItem：删除序列号为i的条目，参数为i；

setItemText：改变第i个位置的条目，参数为i；

insertSeparator：在序号为i的条目前插入分割线，参数为i；

setCurrentIndex：将序号为i的条目设置为默认显示的内容，参数为i；

setEditable：设置为可编辑状态，参数为Bool，增加的项目会放在后面；

setMaxCount：设置可以存储的最大项数；maxCount：获取可以存储的最大项数

setMaxVisibleItems：设置展示的最大项数；maxVisibleItems：获取展示的最大项数

setDuplicatesEnabled：设置是否允许用户添加重复项，参数为Bool；duplicatesEnabled：获取是否可以被重复；

setFrame：设置框架边框，参数为Bool；hasFrame：获取是否有框架边框

setIconSize：设置图标尺寸；iconSize：获取图标尺寸；

setSizeAdjustPolicy：设置调整尺寸策略，（参数AdjustToContents始终根据内容调整；AdjustToContentsOnFirstShow固定在提一次提示时的大小；AdjustToMinimumContentsLength 最小宽度；AdjustToMinimumContentsLengthWithIcon：包含图标的最小宽度）

setCompleter，setValidator：完成器和验证器设置；

activated：当用户选中一个下拉选项时发射该信号

currentIndexChanged：当下拉选项的索引发生改变时发射该信号，当参数为int时，传递的是当前项的索引号，当参数为str时，传递的是当前选择项的字符串；

highlighted：当选中一个已经选中的下拉选项时（只要鼠标放在条目上就触发），发射该信号；

# QComboBox支持模型/视图框架相关操作

#------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractSpinBox

【继承关系】QtWidget. QabstractSpinBox -> QtWidget. QWidget

【继承者】QDateEdit, QDateTimeEdit, QDoubleSpinBox, QSpinBox, QTimeEdit

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  alignment  buttonSymbols  changeEvent  childEvent  clear  closeEvent  connectNotify  contextMenuEvent  correctionMode  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  editingFinished  enterEvent  event | fixup  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hasAcceptableInput  hasFrame  hideEvent  initPainter  initStyleOption  inputMethodEvent  inputMethodQuery  interpretText  isAccelerated  isGroupSeparatorShown  isReadOnly  isSignalConnected  keyboardTracking  keyPressEvent  keyReleaseEvent | leaveEvent  lineEdit  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent  selectAll  sender  senderSignalIndex  setAccelerated  setAlignment  setButtonSymbols  setCorrectionMode  setFrame | setGroupSeparatorShown  setKeyboardTracking  setLineEdit  setReadOnly  setSpecialValueText  setWrapping  sharedPainter  showEvent  sizeHint  specialValueText  stepBy  stepDown  stepEnabled  stepUp  tabletEvent  text  timerEvent  updateMicroFocus  validate  wheelEvent  wrapping |

【详细描述】QAbstractSpinBox是一个抽象类，是所有步长调节器的通用的功能抽象出了一个父类。虽然QAbstractSpinBox是一个抽象类，但是可以直接实例化使用。QAbstractSpinBox包含了一个QLineEdit和两个QPushbutton。数据的更改可以通过点击按钮或使用键盘输入。

text：返回文本框内所有字符；

setButtonSymbols：设置旋转框中按钮上显示的符号，枚举值【UpDownArrows，PlusMinus，NoButtons】；buttonSymbols：获取设置的参数；

setCorrectionMode：设置纠正模式，用户输入非法字符时调用，有恢复前一值及取一个最接近的值，默认前者，枚举值【CorrectToPreviousValue，CorrectToNearestValue】； CorrectionMode：获取纠正模式的参数；

setAccelerated：设置加速开关，默认关。开启时，长按右端小箭头，值的增大速度由慢变快，适合值的大范围变化，参数为bool；isAccelerated：返回是否启用加速开关的判断；

setAlignment：设置对其方式，参数为枚举值，参见Qt.AlignmentFlag对象；

setFrame：设置边框，参数为bool；hasFrame：返回是否启用边框的判断；

setKeyboardTracking：按键追踪，默认true，此时，当用户输入的值变化时，信号valueChanged信号不断发出，false时，失去焦点或点击箭头时才发出；keyboardTracking：返回是否按键追踪；

setReadOnly：设置只读模式，不可编辑，参数为bool；isReadOnly：返回是否启用只读模式的判断；

setSpecialValueText：设置一个特殊值，当设置一个变化值0-100，当设置为0时，不显示0，而显示这个txt；specialValueText：返回特殊值；

setWrapping：设置循环，值增长到最大时，自动跳到最小值，返之亦然，参数为bool； wrapping：返回是否启用循环的判断；

stepBy：设置每一个step，参数为int；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSpinBox，QDoubleSpinBox

【继承关系】QtWidget.QSpinBox，QtWidget.QDoubleSpinBox -> QtWidget.QAbstractSpinBox –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| cleanText  displayIntegerBase  maximum  minimum  prefix  setDisplayIntegerBase | setMaximum  setMinimum  setPrefix  setRange  setSingleStep  setStepType | setSuffix  setValue[Slots]  singlestep  stepType  suffix  textChanged[Sig] | textFromValue  value  valueChanged[Sig]  valueFromText |

【详细描述】

QSpinBox：是专门用于输入输出整数的组件，还可以用于二进制，十六进制显示；

QDoubleSpinBox：是专门用于输入输出浮点数的组件；

【常用函数】：

setPrefix：设置数字显示的前缀，如”$”；prefix：获取前缀；setSuffix：设置数字显示的前缀，如”$”；suffix：获取后缀；

setRange：设置计数器的最大值，最小值，参数为min，max；

setMinimum：设置数字显示的最大值，输入为int；minimum：返回数字显示的最小值，如”0”；

setMaximum：设置数字显示的最大值，输入为int；maximum：返回数字显示的最大值；

setWrapping：设定步长；Singlesetp：获取步长；

value：当前显示的值；

displayIntegerBase：QSpinBox特有属性，显示整数使用的进制，例如2就表示二进制；

decimals：QDoubleSpinBox特有属性，显示数值的小数位数，例如2就表示显示两位小数；

setDisplayIntegerBase：设置进制；displayIntegerBase：获取进制；

setValue：设置值；value：获取值；

textFromValue：用于自定义展示格式，参数为int；

valueChanged：传递有int类型的信号，还可以重构发送字符串类型的数据；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractSlider

【继承关系】QtWidget.QAbstractSlider –> QtWidget.QWidget

【继承者】QDial，QScrollBar，QSlider

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  actionTriggered  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild | focusOutEvent  focusPreviousChild  hasTracking  hideEvent  initPainter  inputMethodEvent  invertedAppearance  invertedControls  isSignalConnected  isSliderDown  keyPressEvent  keyReleaseEvent  leaveEvent  maximum  metric  minimum  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent | moveEvent  nativeEvent  orientation  pageStep  paintEvent  rangeChanged  receivers  repeatAction  resizeEvent  sender  senderSignalIndex  setInvertedAppearance  setInvertedControls  setMaximum  setMinimum  setOrientation  setPageStep  setRange  setRepeatAction  setSingleStep | setSliderDown  setSliderPosition  setTracking  setValue  sharedPainter  showEvent  singleStep  sliderChange  sliderMoved  sliderPosition  sliderPressed  sliderReleased  tabletEvent  timerEvent  triggerAction  updateMicroFocus  value  valueChanged  wheelEvent |

【详细描述】QAbstractSlider()是QWidget()的子类，提供了一个范围内的整数值。它是QSlider,QScrollBar和QDial的父类（抽象类），将多种滑动控件的共性抽象了出来。QAbstractSlider()是个抽象类，使用必须通过子类来实例化。下面的功能演示都有QSlider来实现。

setMinimum：设置滑动块条控件的最小值；setMaximum：设置滑动块条控件的最大值；

setValue：设置前值；value：获取当前值；

setTracking：打开或关闭滑块跟踪，如果启用跟踪(默认)，滑块在拖动滑块时发出valueChanged()信号。如果禁用跟踪，则滑块仅在用户释放滑块时才发出valueChanged()信号。

setOrientation：设置水平滑动条或垂直滑动条，参数为枚举值，参数为Qt.Orientation对象；

sliderMoved：移动滑动条时发出的信号；

sliderPressed：点击滑动条时所发出的信号；

valueChanged：数值改变时所发出的信号：

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSlider

【继承关系】QtWidget QSlider -> QtWidget.QAbstractSlider –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent | enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  initStyleOption  inputMethodEvent  isSignalConnected  keyPressEvent  keyReleaseEvent | leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  repeatAction  resizeEvent  sender | senderSignalIndex  setRepeatAction  setTickInterval  setTickPosition  sharedPainter  showEvent  sizeHint  sliderChange  tabletEvent  tickInterval  tickPosition  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】滑动条控件，提供了一个垂直或水平的滑块。它允许用户沿水平或垂直方向在一定的范围内移动滑块，并将滑块的位置映射为一个定义范围内的数值

从表现形式来看，滑块分为两类（垂直、水平），定义方法：QSlider()，参数为枚举值，参数为Qt.Orientation对象；

setSingleStep：设置在键盘上按下→键或←键时的步进值；setPageStep：设置鼠标点击时移动的步进值；

setTickInterval：设置刻度间隔；

setTickPosition：设置客户标记的位置，参数为枚举值【NoTicks，不会只任何刻度线；TicksAbove/TicksBelow(TicksLeft/ TicksRight)，在（水平或垂直）滑块上/下（左/右）方绘制刻度线；TciksBothSides：在滑块的两侧绘制刻度线】

sliderReleased：释放时所发出的信号；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDialog

【继承关系】QtWidget. QDialog –> QtWidget.QWidget

【继承者】QAbstractPrintDialog，QPageSetupDialog，QPrintDialog，QPrintPreviewDialog，QColorDialog，QErrorMessage，QFileDialog，QFontDialog，QInputDialog，QMessageBox，QProgressDialog，QWizard

|  |  |  |  |
| --- | --- | --- | --- |
| accept  accepted  actionEvent  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  done  dragEnterEvent  dragLeaveEvent  dragMoveEvent | dropEvent  enterEvent  event  eventFilter  exec  exec\_  finished  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  inputMethodEvent  isSignalConnected | isSizeGripEnabled  keyPressEvent  keyReleaseEvent  leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  open  paintEvent  receivers  reject | rejected  resizeEvent  result  sender  senderSignalIndex  setModal  setResult  setSizeGripEnabled  setVisible  sharedPainter  showEvent  sizeHint  tabletEvent  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】QDialog类是所有对话框窗口的基类，对话框窗口是一个经常用来完成短小任务或者和用户进行简单交互的顶层窗口，对话框分为模态对话框和非模态对话框，模态对话框在关闭它之前,不能与同一个应用程序的其他窗口进行交互；非模态对话框既可以和它交互,也可以和同一个应用程序的其他窗口交互；模态对话框用 exec() 函数显示,或者在 show() 函数之前加上setModal(true)，而非模态对话框用 show() 函数显示；

setWindowModality：设置模态级别，参数为枚举值，参数为Qt.WindowModality对象；

setSizeGripEnabled：设置显示尺寸是否可以调整，参数为bool；isSizeGripEnabled：查看；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFileDialog

【继承关系】QtWidget. QFileDialog –> QtWidget. QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| accept  acceptMode  actionEvent  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  currentChanged  currentUrlChanged  customEvent  defaultSuffix  destroy  directory  directoryEntered  directoryUrl  directoryUrlEntered  disconnectNotify  done  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  eventFilter  fileMode  fileSelected | filesSelected  filter  filterSelected  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  getExistingDirectory  getExistingDirectoryUrl  getOpenFileName  getOpenFileNames  getOpenFileUrl  getOpenFileUrls  getSaveFileName  getSaveFileUrl  hideEvent  history  iconProvider  initPainter  inputMethodEvent  isSignalConnected  itemDelegate  keyPressEvent  keyReleaseEvent  labelText  leaveEvent  metric  mimeTypeFilters | mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nameFilters  nativeEvent  open  options  paintEvent  proxyModel  receivers  resizeEvent  restoreState  saveFileContent  saveState  selectedFiles  selectedMimeTypeFilter  selectedNameFilter  selectedUrls  selectFile  selectMimeTypeFilter  selectNameFilter  selectUrl  sender  senderSignalIndex  setAcceptMode  setDefaultSuffix  setDirectory | setDirectoryUrl  setFileMode  setFilter  setHistory  setIconProvider  setItemDelegate  setLabelText  setMimeTypeFilters  setNameFilter  setNameFilters  setOption  setOptions  setProxyModel  setSidebarUrls  setSupportedSchemes  setViewMode  setVisible  sharedPainter  showEvent  sidebarUrls  supportedSchemes  tabletEvent  testOption  timerEvent  updateMicroFocus  urlSelected  urlsSelected  viewMode  wheelEvent |

【详细描述】用于打开和保存文件的标准对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QFileDialog的标题（str）；3. 弹窗的初始化路径，如果没有设置的话，则是当前程序运行的路径作为弹窗的打开路径（dir）；4. 过滤条件’Images(\*.png \*.gif);;Text files(\*.txt);;XML files(\*.xml)’；如何运行对话框的选项（options，QFileDialog.Option对象，1. showDirsOnly，在对话框中只显示目录，默认时显示目录和文件夹；2. DontConfirmOverWrite，覆盖一个已经存在的文件时不进行提示，默认是要提示的；3. DontUseNoativeDialog，不使用本地对话框，默认时使用本地对话框的；ReadOnly，对话框显示的文件系统是只读的）】，返回值为文本编辑框内的字符串；

getOpenFileName：返回用户所选择的名称，并打开该文件，参数与QFileDialog相同getOpenFileNames：返回用户所选择的名称，并打开多个文件，参数与QFileDialog相同；

getOpenFileUrl：获取一个打开文件的统一资源定位符，参数与QFileDialog相同；getOpenFileUrls：获取多个打开文件的统一资源定位符，参数与QFileDialog相同；

getExistingDirectory：打开文件夹，参数与QFileDialog相同；

getSaveFileName：获取用户选择的文件名并保存文件，参数与QFileDialog相同；

getSaveFileUrl：获取保存的统一资源定位符，参数与QFileDialog相同；

setFileMode：可以选择对话框返回的文件类型，参数为枚举值【AnyFile，任何文件；ExistingFile，已存在的文件；Directory，文件目录；ExistingFiles，已经存在的多个文件】；

setViewMode：设置对话框显示信息的详细程度，参数为枚举值【Detail，详细；List，仅列表】，但在Win10中没有效果；

setAcceptMode：设置是打开还是保存文件对话框，参数为枚举值【AcceptOpen，文件对话框为打开类型；AcceptSave，文件对话框为保存类型】

setDefaultSuffix：设置默认后缀，参数为str；

getExistingDirectory：获取文件夹；

getExistingDirectoryUrl：获取文件夹的统一资源定位符；

directory：获取对话框中的目录地址，返回QDir对象；

setFilter：设置过滤器，只显示过滤器允许的文件类型，参数为枚举值，【QDir.filter对象】；

setHistory：设置历史目录；

currentChanged/currentUrlChanged：当前变化时发信号（只要鼠标点中文件或文件夹就触发）；

directoryEntered/directoryUrlEntered：进入目录时触发信号；

fileSelected：选择时触发信号；

filterSelected：名称字符串过滤器变化时触发信号；

exec：显示一个模式对话框，并且锁住程序直到用户关闭该对话框为止，函数返回一个DialogCode结果。在对话框弹出期间，用户不可以切换同程序下的其它窗口，直到该对话框被关闭。

show：显示一个非模式对话框。控制权即刻返回给调用函数，弹出窗口是否模式对话框，取决于modal属性的值。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QColorDialog

【继承关系】QtWidget. QColorDialog –> QtWidget. QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  changeEvent  childEvent  closeEvent  colorSelected  connectNotify  contextMenuEvent  create  currentColor  currentColorChanged  customColor  customCount  customEvent  destroy  disconnectNotify  done  dragEnterEvent | dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  getColor  hideEvent  initPainter  inputMethodEvent  isSignalConnected  keyPressEvent | keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  open  options  paintEvent  receivers  resizeEvent  selectedColor  sender  senderSignalIndex | setCurrentColor  setCustomColor  setOption  setOptions  setStandardColor  setVisible  sharedPainter  showEvent  standardColor  tabletEvent  testOption  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】选择颜色对话框；

getColor：打开颜色；

#-------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFontDialog

【继承关系】QtWidget.QFontDialog –> QtWidget. QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  currentFont  currentFontChanged  customEvent  destroy  disconnectNotify  done  dragEnterEvent  dragLeaveEvent | dragMoveEvent  dropEvent  enterEvent  event  eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  fontSelected  getFont  hideEvent  initPainter  inputMethodEvent | isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  open  options  paintEvent  receivers | resizeEvent  selectedFont  sender  senderSignalIndex  setCurrentFont  setOption  setOptions  setVisible  sharedPainter  showEvent  tabletEvent  testOption  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】字体选择对话框，可以显示文本的字号大小，样式和格式；

getFont：执行模式字体对话框并返回字体，并通过setFont进行设定；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMessageBox

【继承关系】QtWidget. QMessageBox –> QtWidget. QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| about  aboutQt  actionEvent  addButton  button  ；buttonClicked  buttonRole  buttons  changeEvent  checkBox  childEvent  clickedButton  closeEvent  connectNotify  contextMenuEvent  create  critical  customEvent  defaultButton  destroy  detailedText  disconnectNotify | dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  escapeButton  event  eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  icon  iconPixmap  information  informativeText  initPainter  inputMethodEvent  isSignalConnected  keyPressEvent | keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  open  paintEvent  question  receivers  removeButton  resizeEvent  sender  senderSignalIndex  setCheckBox  setDefaultButton  setDetailedText  setEscapeButton  setIcon | setIconPixmap  setInformativeText  setStandardButtons  setText  setTextFormat  setTextInteractionFlags  setWindowModality  setWindowTitle  sharedPainter  showEvent  standardButton  standardButtons  standardIcon  tabletEvent  text  textFormat  textInteractionFlags  timerEvent  updateMicroFocus  warning  wheelEvent |

【详细描述】QMessageBox是一种通用的弹出式对话框，用于显示消息，允许用户通过单击不同的标准按钮对消息进行反馈，常用的弹出式对话框，如提示、警告、错误、问询、关于等对话框，QMessageBox内置的常用按钮：NoButton；Ok；Cancel；Yes；No；About；Retry；Ingnore；YesAll；NoAll；Close；Discard；

question：弹出问答对话框，参数包含【parent，指定的父窗口控件；title，对话框标题；text，对话框文本；buttons，多个标准按钮，默认为ok按钮；defaulButton，默认选中的标准按钮，默认是第一个标准按钮】；

warning：弹出警告对话框，同上；

ctitical：弹出严重错误对话框，同上；

about：弹出关于对话框，同上；

setTitle：设置标题；setText：设置消息正文；setIcon：设置弹出对话框的图片；

setWindowModality：设置模态级别，参数为枚举值，参数为Qt.WindowModality对象；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QInputDialog

【继承关系】QtWidget.QInputDialog –> QtWidget.QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  cancelButtonText  changeEvent  childEvent  closeEvent  comboBoxItems  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  done  doubleDecimals  doubleMaximum  doubleMinimum  doubleStep  doubleValue  doubleValueChanged  doubleValueSelected  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event | eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  getDouble  getInt  getItem  getMultiLineText  getText  hideEvent  initPainter  inputMethodEvent  inputMode  intMaximum  intMinimum  intStep  intValue  intValueChanged  intValueSelected  isComboBoxEditable  isSignalConnected  keyPressEvent  keyReleaseEvent  labelText | leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  okButtonText  open  options  paintEvent  receivers  resizeEvent  sender  senderSignalIndex  setCancelButtonText  setComboBoxEditable  setComboBoxItems  setDoubleDecimals  setDoubleMaximum  setDoubleMinimum  setDoubleRange  setDoubleStep  setDoubleValue | setInputMode  setIntMaximum  setIntMinimum  setIntRange  setIntStep  setIntValue  setLabelText  setOkButtonText  setOption  setOptions  setTextEchoMode  setTextValue  setVisible  sharedPainter  showEvent  sizeHint  tabletEvent  testOption  textEchoMode  textValue  textValueChanged  textValueSelected  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】QInputDialog控件是一个标准对话框，由一个文本框和两个按钮（OK按钮和Cannel按钮）组成。其包含多种输入方式，a. 单行字符串输入；b. 整数输入；c. 浮点数输入；d. 列表框选择输入；e. 多行文本；

getInt：整数输入对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QInputDialog的标题（str）；3. QInputDialog内部显示的文本（str）；4. Spin Box控件默认值（value）；5. Spin Box控件最小值（min）；6. Spin Box控件最大值（max）；7. Spin Box的步进值（step）；8. 通过选择不同的inputMethodHints值来实现不同的键盘布局（inputMethodHints）】，返回值为编辑框内的）整数值；

getDouble：浮点数输入对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QInputDialog的标题（str）；3. QInputDialog内部显示的文本（str）；4. Spin Box控件默认值（value）；5. Spin Box控件最小值（min）；6. Spin Box控件最大值（max）；7.Spin Box控件显示的小数点位数控制（decimals）；8. 通过选择不同的inputMethodHints值来实现不同的键盘布局（inputMethodHints）】，返回值为Double SpinBox控件的浮点值；

getItem：下拉列表选择输入对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QInputDialog的标题（str）；3. QInputDialog内部显示的文本（str）；4. ComboBox组件的内容列表（list-of-str）；5. 默认显示ComboBox组件哪一个索引的内容（int）；6.ComboBox组件是否可被编辑（editable）；7. 指明标准输入对话框的窗体标识（flags）；8. 通过选择不同的inputMethodHints值来实现不同的键盘布局（inputMethodHints）】，返回值为文本编辑框内的字符串；

getText：单行字符串输入对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QInputDialog的标题（str）；3. QInputDialog内部显示的文本（str）；4.文本编辑框内容的显示方式（echo mode）；5.文本编辑框默认显示内容（str）；6. 指明标准输入对话框的窗体标识（flags）；7. 通过选择不同的inputMethodHints值来实现不同的键盘布局（inputMethodHints）】，返回值为文本编辑框内的字符串；

getMultiLineText：多行文本对话框，其参数包含【1. 指定的父窗口控件（parent）；2. QInputDialog的标题（str）；3. QInputDialog内部显示的文本（str）；4.文本编辑框内容的显示方式（echo mode）；5.文本编辑框默认显示内容（str）；6. 指明标准输入对话框的窗体标识（flags）；7. 通过选择不同的inputMethodHints值来实现不同的键盘布局（inputMethodHints）】，返回值为文本编辑框内的字符串；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPainter

【继承关系】QtGui.QPainter

【继承者】QStylePainter

|  |  |  |  |
| --- | --- | --- | --- |
| background  backgroundMode  begin  beginNativePainting  boundingRect  brush  brushOrigin  clipBoundingRect  clipPath  clipRegion  combinedTransform  compositionMode  device  deviceTransform  drawArc  drawChord  drawConvexPolygon  drawEllipse  drawGlyphRun  drawImage  drawLine  drawLines | drawPath  drawPicture  drawPie  drawPixmap  drawPixmapFragments  drawPoint  drawPoints  drawPolygon  drawPolyline  drawRect  drawRects  drawRoundedRect  drawStaticText  drawText  drawTiledPixmap  end  endNativePainting  eraseRect  fillPath  fillRect  font  fontInfo | fontMetrics  hasClipping  isActive  layoutDirection  opacity  paintEngine  pen  renderHints  resetTransform  restore  rotate  save  scale  setBackground  setBackgroundMode  setBrush  setBrushOrigin  setClipPath  setClipping  setClipRect  setClipRegion  setCompositionMode | setFont  setLayoutDirection  setOpacity  setPen  setRenderHint  setRenderHints  setTransform  setViewport  setViewTransformEnabled  setWindow  setWorldMatrixEnabled  setWorldTransform  shear  strokePath  testRenderHint  transform  translate  viewport  viewTransformEnabled  window  worldMatrixEnabled  worldTransform |

【详细描述】QPainter是用来进行绘图操作的类，一般的绘图设备包括QWidget，QPixmap，QImage等，而这些绘图设备为QPainter提供了一个画布。

QPainter提供高度优化的功能来完成大多数图形用户界面程序所需的工作。它能画出从简单线条到复杂形状如饼图和弦等一切图形。它还可以绘制对齐的文本和像素图。通常，它绘制一个“自然”坐标系，但它也可以进行视图和世界变换。QPainter可以对继承QPaintDevice类的任何对象进行操作。

QPainter的绘制操作在QWidget.paintEvent()中完成，绘制方法必须放在QtGui.QPainter对象的begin（）和end（）之间。QPainter类在控件或其他绘图设备上执行较低级别的图像绘制功能，并通过如表的方法进行绘制；

QPainter类的成员角色有：

QPen：用于绘制几何图形的边缘,由颜色,宽度,线风格等参数组成

QBrush：用于填充几何图形的调色板,由颜色和填充风格组成

QFont：用于文本绘制

QPixmap：绘制图片,可以加速显示,带有屏幕截图,窗口截图等支持,适合小图片

QImage：绘制图片,可以直接读取图像文件进行像素访问,适合大图片

QBitmap：QPixmap的一个子类，主要用于显示单色位图

QPicture：绘图装置，用于记录和重播Qpainter的绘图指令

【函数】

begin/end：开始/结束在目标设备上绘制；

drawArc：在起始角度和最终角度之间画弧；

drawEllipse：在一个矩形内画一个椭圆；

drawLine：绘制一条指定了端点坐标的线，绘制从（x1, y1）到（x2,y2）的直线，并且设置画笔位置为（x2,y2）；

drawPixmap：从图像文件中提取Pixmap并将其显示在指定的位置，参数1. int, 起始坐标；2. int, 起始坐标；3. width, 宽度；4. height，高度；5. QPixmap对象；

drawPolygon：使用坐标数组绘制多边形；

drawRect：以给定的宽度和高度从左上角坐标（x, y）绘制一个矩形，参数分别为（x, y, w, h）；

drawText：显示给定坐标出的文字；

fillRect：使用QColor参数填充矩形；

setBrush：设置画笔填充的风格；

setPen：设置用于绘制的笔的颜色，大小和样式；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QProgressDialog

【继承关系】QtWidget. QProgressDialog –> QtWidget.QDialog –> QtWidget.QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  autoClose  autoReset  cancel  canceled  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent | event  eventFilter  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  forceShow  hideEvent  initPainter  inputMethodEvent  isSignalConnected  keyPressEvent  keyReleaseEvent  labelText  leaveEvent  maximum  metric  minimum | minimumDuration  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  open  paintEvent  receivers  reset  resizeEvent  sender  senderSignalIndex  setAutoClose  setAutoReset  setBar  setCancelButton  setCancelButtonText | setLabel  setLabelText  setMaximum  setMinimum  setMinimumDuration  setRange  setValue  sharedPainter  showEvent  sizeHint  tabletEvent  timerEvent  updateMicroFocus  value  wasCanceled  wheelEvent |

【详细描述】设置显示进度的对话框，可以在大的循环操作中显示操作的进度；

setWindowModality：设置模态级别，参数为枚举值，参数为Qt.WindowModality对象；

setAutoReset：设置当Value达到最大值时自动调用reset函数，参数为Bool；reset：使进度对话框复位，如果autoClose属性为True，将隐藏对话框；

sertAutoClose：调用reset函数时隐藏窗口；

setValue：为对话框上的进度条设置一个整数值Progress，设置后进度条会自动刷新显示；

cancel：使对话框复位，并且wasCanceled函数返回True；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QImage

【继承关系】QtGui.QImage -> QtGui.QPaintDevice

|  |  |  |  |
| --- | --- | --- | --- |
| allGray  applyColorTransform  bitPlaneCount  bits  byteCount  bytesPerLine  cacheKey  color  colorCount  colorSpace  colorTable  constBits  constScanLine  convertedToColorSpace  convertTo  convertToColorSpace  convertToFormat  copy  createAlphaMask  createHeuristicMask | createMaskFromColor  depth  detach  devicePixelRatio  devType  dotsPerMeterX  dotsPerMeterY  fill  format  fromData  hasAlphaChannel  height  invertPixels  isDetached  isGrayscale  isNull  load  loadFromData  metric  mirrored | offset  paintEngine  pixel  pixelColor  pixelFormat  pixelIndex  rect  reinterpretAsFormat  rgbSwapped  save  scaled  scaledToHeight  scaledToWidth  scanLine  setColor  setColorCount  setColorSpace  setColorTable  setDevicePixelRatio  setDotsPerMeterX | setDotsPerMeterY  setOffset  setPixel  setPixelColor  setText  size  sizeInBytes  smoothScaled  swap  text  textKeys  toImageFormat  toPixelFormat  transformed  trueMatrix  valid  width |

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPixmap

【继承关系】QtGui.QPixmap -> QtGui.QPaintDevice

【继承者】QBitmap

|  |  |  |  |
| --- | --- | --- | --- |
| cacheKey  convertFromImage  copy  createHeuristicMask  createMaskFromColor  defaultDepth  depth  detach  devicePixelRatio  devType | fill  fromImage  fromImageReader  hasAlpha  hasAlphaChannel  height  isNull  isQBitmap  load  loadFromData | mask  metric  paintEngine  rect  save  scaled  scaledToHeight  scaledToWidth  scroll  setDevicePixelRatio | setMask  size  swap  toImage  transformed  trueMatrix  width |

【详细描述】QPixmap用于绘图设备的图像显示，它可以作为一个QPaintDevice对象，也可以加载到一个控件中，通常是标签或按钮，用于在标签或按钮上显示图像；Pixmap类可以读取的图像文件类型有BMP、GIF、JPG、JPEG、PNG、PBM、PGM、PPM、XBM、XPM等，基本涵盖了我们平时经常用到的图片格式。构造函数：QPixmap(filename, format, flags)，参数filename 文件名；format 字符串，表示图像文件的格式，如果为 0，将进行自动识别；flags 表示颜色的转换模式【Qt.AutoColor：由系统自动决定；Qt.ColorOnly：彩色模式；Qt.MonoOnly：单色模式】。

loadFromData：从给定数据data中加载图像

copy：从QRect对象复制到QPixmap对象；fromImage：将QImage对象转换为QPixmap对象；

grabWidget：从给定的窗口小控件创建一个像素图；grabWindow：在窗口中创建数据的像素图；

load：加载图像文件作为QPixmap对象；

save：将QPixmap对象保存为文件；toImage：将QPixmap对象转换为Image对象；

depth：获取颜色深度，既每像素所占的比特数；width：图像宽度，单位是像素；height：像高度，单位是像素；

size：图像的大小，即 QSize(width(), height())；

isNull：判断是否为空图像；

scaledToWidth：返回一个缩放后的图片的QPixmap对象副本，并将图片的宽度缩放到width，参数width[int]；

scaledToHeight：返回一个缩放后的图片的QPixmap对象副本，并将图片的高度缩放到height，参数height[int]；

scaled：返回一个缩放后的图片的副本，图片缩放到宽度width和高度height，缺省为不保持比例，参数1. width；2. height；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QBitmap

【继承关系】QtGui.QBitmap -> QtGui.QPixmap -> QtGui.QPaintDevice

|  |  |  |  |
| --- | --- | --- | --- |
| clear  fromData | fromImage  metric | swap  transformed |  |

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPicture

【继承关系】QtGui.QPicture -> QtGui.QPaintDevice

|  |  |  |  |
| --- | --- | --- | --- |
| boundingRect  data  detach  devType | isDetached  isNull  load  metric | paintEngine  play  save  setBoundingRect | setData  size  swap |

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QCalendar

【继承关系】QtGui. QCalendar

|  |  |  |  |
| --- | --- | --- | --- |
| availableCalendars  dateFromParts  dateFromParts  dateTimeToString  dayOfWeek  daysInMonth  daysInYear | hasYearZero  isDateValid  isGregorian  isLeapYear  isLunar  isLuniSolar  isProleptic | isSolar  isValid  maximumDaysInMonth  maximumMonthsInYear  minimumDaysInMonth  monthName  monthsInYear | name  partsFromDate  standaloneMonthName  standaloneWeekDayName  weekDayName |

【详细描述】QCalendar是一个日历控件，它提供了一个基于月份的视图，允许用户通过鼠标或者日期，默认选中的是今天的日期。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QCalendarWidget

【继承关系】QtWidgets. QCalendar –> QtWidgets. QtWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  activated  calendar  changeEvent  childEvent  clicked  closeEvent  connectNotify  contextMenuEvent  create  currentPageChanged  customEvent  dateEditAcceptDelay  dateTextFormat  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  eventFilter  firstDayOfWeek | focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  headerTextFormat  hideEvent  horizontalHeaderFormat  initPainter  inputMethodEvent  isDateEditEnabled  isGridVisible  isNavigationBarVisible  isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  maximumDate  metric  minimumDate  minimumSizeHint  monthShown  mouseDoubleClickEvent  mouseMoveEvent | mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintCell  paintEvent  receivers  resizeEvent  selectedDate  selectionChanged  selectionModel  sender  senderSignalIndex  setCalendar  setCurrentPage  setDateEditAcceptDelay  setDateEditEnabled  setDateRange  setDateTextFormat  setFirstDayOfWeek  setGridVisible  setHeaderTextFormat  setHorizontalHeaderFormat  setMaximumDate | setMinimumDate  setNavigationBarVisible  setSelectedDate  setSelectionModel  setVerticalHeaderFormat  setWeekdayTextFormat  sharedPainter  showEvent  showNextMonth  showNextYear  showPreviousMonth  showPreviousYear  showSelectedDate  showToday  sizeHint  tabletEvent  timerEvent  updateCell  updateCells  updateMicroFocus  verticalHeaderFormat  weekdayTextFormat  wheelEvent  yearShown |

【详细描述】

setDataRange：设置日期范围供选择；

setFirstDayOfWeek：重新设置星期的第一天，默认是星期日。参数为枚举值，【Qt.DayOfWeek对象】；

setMinimumDate：设置最小日期；MinimumDate：获取最小值；setMaximumDate：设置最大日期；MaximumDate：获取最大值；

setSelectedDate：设置一个QDate对象，作为日期控件所选定的日期；SelectedDate：返回当前选的日期；

setGridvisible：设置日历控件是否显示网格；

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDateTimeEdit

【继承关系】QtWidgets.QDateTimeEdit –> QtWidgets. QAbstractSpinBox -> QtWidgets. QtWidget

【继承者】QDateEdit, QTimeEdit

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  calendar  calendarPopup  calendarWidget  changeEvent  childEvent  clear  clearMaximumDate  clearMaximumDateTime  clearMaximumTime  clearMinimumDate  clearMinimumDateTime  clearMinimumTime  closeEvent  connectNotify  contextMenuEvent  create  currentSection  currentSectionIndex  customEvent  date  dateChanged  dateTime  dateTimeChanged  dateTimeFromText  destroy  disconnectNotify | displayedSections  displayFormat  dragEnterEvent  dragLeaveEvent  dragMoveEvent  dropEvent  enterEvent  event  fixup  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  initStyleOption  inputMethodEvent  isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  lineEdit  maximumDate  maximumDateTime  maximumTime  metric | minimumDate  minimumDateTime  minimumTime  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent  sectionAt  sectionCount  sectionText  sender  senderSignalIndex  setCalendar  setCalendarPopup  setCalendarWidget  setCurrentSection  setCurrentSectionIndex  setDate  setDateRange  setDateTime  setDateTimeRange  setDisplayFormat | setLineEdit  setMaximumDate  setMaximumDateTime  setMaximumTime  setMinimumDate  setMinimumDateTime  setMinimumTime  setSelectedSection  setTime  setTimeRange  setTimeSpec  sharedPainter  showEvent  sizeHint  stepBy  stepEnabled  tabletEvent  textFromDateTime  time  timeChanged  timerEvent  timeSpec  updateMicroFocus  validate  wheelEvent |

【详细描述】QDateTimeEdit是一个允许用户编辑日期时间的控件，可以使用键盘和上，下箭头按钮来增加或减少日期时间值。

setDisplayFormat：设置日期时间格式，其中yyyy表示年，MM表示月，dd表示日，HH表示小时，mm表示分钟，ss表示秒；

setMinimumDate：设置空间的最小日期；setMinimumDate：设置控件的最大日期；

time：返回编辑的时间；date：返回编辑的日期；

setCalendarPopup：设置日历控件允许弹出，参数为Bool

dateChanged：当日期发生改变时，触发此信号；

dateTimeChanged：当日期时间发生改变时，触发此信号；

timeChanged：当时间发生改变时，触发此信号；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractItemView

【继承关系】QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

【继承者】QHelpContentWidget，QHelpIndexWidget，QColumnView，QHeaderView，QListView，QListWidget，QTableView，QTableWidget，QTreeView，QTreeWidget，QUndoView

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  activated  alternatingRowColors  autoScrollMargin  changeEvent  childEvent  clearSelection  clicked  closeEditor  closeEvent  closePersistentEditor  commitData  connectNotify  contextMenuEvent  create  currentChanged  currentIndex  customEvent  dataChanged  defaultDropAction  destroy  dirtyRegionOffset  disconnectNotify  doubleClicked  dragDropMode  dragDropOverwriteMode  dragEnabled  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  edit  editorDestroyed  editTriggers  entered  enterEvent  event  eventFilter  executeDelayedItemsLayout | focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hasAutoScroll  hideEvent  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  horizontalScrollMode  iconSize  iconSizeChanged  indexAt  indexWidget  initPainter  initStyleOption  inputMethodEvent  inputMethodQuery  isIndexHidden  isPersistentEditorOpen  isSignalConnected  itemDelegate  itemDelegateForColumn  itemDelegateForRow  keyboardSearch  keyPressEvent  keyReleaseEvent  leaveEvent  metric  model  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  openPersistentEditor  paintEvent  pressed | receivers  reset  resetHorizontalScrollMode  resetVerticalScrollMode  resizeEvent  rootIndex  rowsAboutToBeRemoved  rowsInserted  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollTo  scrollToBottom  scrollToTop  selectAll  selectedIndexes  selectionBehavior  selectionChanged  selectionCommand  selectionModel  sender  senderSignalIndex  setAlternatingRowColors  setAutoScroll  setAutoScrollMargin  setCurrentIndex  setDefaultDropAction  setDirtyRegion  setDragDropMode  setDragDropOverwriteMode  setDragEnabled  setDropIndicatorShown  setEditTriggers  setHorizontalScrollMode  setIconSize  setIndexWidget  setItemDelegate  setItemDelegateForColumn  setItemDelegateForRow  setModel  setRootIndex | setSelection  setSelectionBehavior  setSelectionModel  setState  setTabKeyNavigation  setTextElideMode  setVerticalScrollMode  setViewportMargins  sharedPainter  showDropIndicator  showEvent  sizeHintForColumn  sizeHintForIndex  sizeHintForRow  startDrag  state  tabKeyNavigation  tabletEvent  textElideMode  timerEvent  update  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  verticalScrollMode  viewOptions  viewportEntered  viewportEvent  viewportMargins  viewportSizeHint  visualRect  visualRegionForSelection  wheelEvent |

【详细描述】QAbstractItemView是所有的使用QAbstractItemModel模型的视图的基类，是一个不能被实例化的抽象类。它通过信号槽机制为与模型的交互操作提供了一个标准化的接口，确保子类视图能够随着模型的变化而及时更新。

setModel：设置View所关联的Model，参数为Model对象；

setSelectionModel：设置选择模型，参数为QItemSelectionModel对象；selectionModel：获取当前的QItemSelectionModel对象，QItemSelectionModel对象用于控制view中操作选择数据项的模式，用于控制用户是否可以选择一个或多个数据项，以及在多个数据项选择中，选择是否必须是一个连续的范围。

setSelectionMode：设置选择模式，参数为枚举值，QAbstractItemView.SelectionMode对象【1. SingleSelection，设置单选模式；2.ContiguousSelection；3. ExtendedSelection，设置多选模式，支持Ctrl，Shift, Ctrl+A，鼠标左键按下滑动多选；4. MultiSelection；5. NoSelection】

setSelectionBehavior：设置单元格的选择方式，参数为枚举值，QAbstractItemView.SelectionBehavior对象【SelectItems，选中单个单元格；SelectRows，选中一行；SelectColumns，选中一列】

setAlternatingRowColors：设置表格的行是否用交替底色显示，若为交替底色，则间隔的一行会用灰色底色，底色的颜色可以用样式表设置；

setEditTriggers：设置视图的项是否可以编辑以及如何进入编辑状态，参数为枚举值，QAbstractItemView.EditTrigger对象【1. NoEditTriggers；2. CurrentChanged；3. DoubleClicked；4. SelectedClicked；5. EditKeyPressed；6. AnyKeyPressed；7. AllEditTriggers】

currentIndex：返回当前项的模型索引QModelIndex；

setItemDelegateForColumn：为某一列设置代理组件，参数1. column->int；2. delegate，代理组件；setItemDelegateForRow：为某一行设置代理组件，参数1. row->int；2. delegate，代理组件；setItemDelegate：为这个视图组件设置代理组件，参数为delegate类型的代理组件；

setDragDropMode：设置对象的DragDropMode属性，参数为QAbstractItemView.dragDropMode对象【1. NoDragDrop，组件不支持拖和放下操作；2. DragOnly,组件只支持拖动操作；3. Drop，组件只接受放下操作；4，DragDrop，组件支持拖动和放下操作；5. InternalMove，组件只支持内部项的移动操作，例如目录树自身节点的移动操作】；dragDropMode：获取DragDropMode的属性值，返回值为bool类型；

setDragEnabled：设置QWidget对象A可以拖动，参数为Bool；

defaultDropAction：返回用于控制QAbstractItemView及其子类的实例视图中拖放时放下的数据操作模式，返回Qt.DropAction对象

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QHeaderView

【继承关系】QtWidgets. QHeaderView –> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  cascadingSectionResizes  changeEvent  childEvent  closeEditor  closeEvent  commitData  connectNotify  contextMenuEvent  count  create  currentChanged  customEvent  dataChanged  defaultAlignment  defaultSectionSize  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  edit  editorDestroyed  enterEvent  event  eventFilter  executeDelayedItemsLayout  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  geometriesChanged  headerDataChanged  hiddenSectionCount  hideEvent  hideSection  highlightSections  horizontalOffset | horizontalScrollbarAction  horizontalScrollbarValueChanged  indexAt  initialize  initializeSections  initPainter  initStyleOption  inputMethodEvent  isFirstSectionMovable  isIndexHidden  isSectionHidden  isSignalConnected  isSortIndicatorShown  keyPressEvent  keyReleaseEvent  leaveEvent  length  logicalIndex  logicalIndexAt  maximumSectionSize  metric  minimumSectionSize  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  moveSection  nativeEvent  offset  orientation  paintEvent  paintSection  receivers  reset  resetDefaultSectionSize  resizeContentsPrecision  resizeEvent  resizeSection  resizeSections  restoreState  rowsAboutToBeRemoved | rowsInserted  saveState  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollTo  sectionClicked  sectionCountChanged  sectionDoubleClicked  sectionEntered  sectionHandleDoubleClicked  sectionMoved  sectionPosition  sectionPressed  sectionResized  sectionResizeMode  sectionsAboutToBeRemoved  sectionsClickable  sectionsHidden  sectionsInserted  sectionSize  sectionSizeFromContents  sectionSizeHint  sectionsMovable  sectionsMoved  sectionViewportPosition  selectedIndexes  selectionChanged  selectionCommand  sender  senderSignalIndex  setCascadingSectionResizes  setDefaultAlignment  setDefaultSectionSize  setDirtyRegion  setFirstSectionMovable  setHighlightSections  setMaximumSectionSize  setMinimumSectionSize  setModel  setOffset  setOffsetToLastSection  setOffsetToSectionPosition | setResizeContentsPrecision  setSectionHidden  setSectionResizeMode  setSectionsClickable  setSectionsMovable  setSelection  setSortIndicator  setSortIndicatorShown  setState  setStretchLastSection  setViewportMargins  setVisible  sharedPainter  showEvent  showSection  sizeHint  sortIndicatorChanged  sortIndicatorOrder  sortIndicatorSection  startDrag  state  stretchLastSection  stretchSectionCount  swapSections  tabletEvent  timerEvent  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  updateSection  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualIndex  visualIndexAt  visualRect  visualRegionForSelection  wheelEvent |

【详细描述】

setStretchLastSection：可以让最后一列填充整个表头；

setSectionResizeMode：可是使表头均分填充表头，参数为QHeaderView.Stretch

setDefaultSectionSize：设置默认的QHeader间距；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTableView

【继承关系】QtWidgets.QTableView –> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

【继承者】QTableWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  changeEvent  childEvent  clearSpans  closeEditor  closeEvent  columnAt  columnCountChanged  columnMoved  columnResized  columnSpan  columnViewportPosition  columnWidth  commitData  connectNotify  contextMenuEvent  create  currentChanged  customEvent  dataChanged  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  edit  editorDestroyed  enterEvent  event  eventFilter  executeDelayedItemsLayout  focusInEvent | focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  gridStyle  hideColumn  hideEvent  hideRow  horizontalHeader  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indexAt  initPainter  initStyleOption  inputMethodEvent  isColumnHidden  isCornerButtonEnabled  isIndexHidden  isRowHidden  isSignalConnected  isSortingEnabled  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  paintEvent  receivers  resizeColumnsToContents | resizeColumnToContents  resizeEvent  resizeRowsToContents  resizeRowToContents  rowAt  rowCountChanged  rowHeight  rowMoved  rowResized  rowsAboutToBeRemoved  rowsInserted  rowSpan  rowViewportPosition  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollTo  selectColumn  selectedIndexes  selectionChanged  selectionCommand  selectRow  sender  senderSignalIndex  setColumnHidden  setColumnWidth  setCornerButtonEnabled  setDirtyRegion  setGridStyle  setHorizontalHeader  setModel  setRootIndex  setRowHeight  setRowHidden  setSelection  setSelectionModel | setShowGrid  setSortingEnabled  setSpan  setState  setVerticalHeader  setViewportMargins  setWordWrap  sharedPainter  showColumn  showEvent  showGrid  showRow  sizeHintForColumn  sizeHintForRow  sortByColumn  startDrag  state  tabletEvent  timerEvent  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalHeader  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualRect  visualRegionForSelection  wheelEvent  wordWrap |

【详细描述】用于显示表格状数据，适用于二维表格型数据的操作；

verticalHeader：返回QTableView控件的QHeaderView对象，以便使用setVisible函数藏列表头，参数为Bool；setHorizontalHeader，对QHeaderView对象进行设置；

horizontalHeader：返回QTableView控件的QHeaderView对象，以便使用setVisible函数藏行表头，参数为Bool；setVerticalHeader，对QHeaderView对象进行设置；

setSpan：合并单元格，参数为(int, int, int, int)，分别为要改变单元格的第row行，第column列，要合并rowSpanCount行数和columnSpanCount列数；

clearSpans：清除所有合并的单元格；

columnAt：根据输入的横坐标（x\_coordinate）位置返回坐标的位置是第几列；

setShowGrid：设置表格是否显示网格线，参数为Bool；

setColumnWidth：设置单元格的宽度；setRowHeight：设置单元格列的高度；

resizeColumnToContents/ resizeRowToContents：自动调整所有列的宽度，以适应其内容，参数如果为Int, 则表示仅调整某一列/行的宽度；

setCornerButtonEnabled：设置表格左上角是否设置勾选框，用于全选所有的单元格，参数为Bool；

setSortingEnabled：设置是否支持排序，参数为Bool；

setModel：设置数据模型；

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

columnCountChanged：

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTableWidget

【继承关系】QtWidgets.QTableWidget -> QtWidgets.QTableView –> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  cellActivated  cellChanged  cellClicked  cellDoubleClicked  cellEntered  cellPressed  cellWidget  changeEvent  childEvent  clear  clearContents  closeEditor  closeEvent  closePersistentEditor  column  columnCount  columnCountChanged  columnMoved  columnResized  commitData  connectNotify  contextMenuEvent  create  currentCellChanged  currentChanged  currentColumn  currentItem  currentItemChanged  currentRow  customEvent  dataChanged  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  dropMimeData | edit  editItem  editorDestroyed  enterEvent  event  eventFilter  executeDelayedItemsLayout  findItems  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  horizontalHeaderItem  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indexFromItem  initPainter  initStyleOption  inputMethodEvent  insertColumn  insertRow  isIndexHidden  isPersistentEditorOpen  isSignalConnected  isSortingEnabled  item  itemActivated  itemAt  itemChanged  itemClicked  itemDoubleClicked  itemEntered  itemFromIndex  itemPressed  itemPrototype  items  itemSelectionChanged  keyPressEvent  keyReleaseEvent | leaveEvent  metric  mimeData  mimeTypes  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  openPersistentEditor  paintEvent  receivers  removeCellWidget  removeColumn  removeRow  resizeEvent  row  rowCount  rowCountChanged  rowMoved  rowResized  rowsAboutToBeRemoved  rowsInserted  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollToItem  selectedIndexes  selectedItems  selectedRanges  selectionChanged  selectionCommand  sender  senderSignalIndex  setCellWidget  setColumnCount  setCurrentCell  setCurrentItem  setDirtyRegion  setHorizontalHeaderItem | setHorizontalHeaderLabels  setItem  setItemPrototype  setModel  setRangeSelected  setRowCount  setSelection  setSortingEnabled  setState  setVerticalHeaderItem  setVerticalHeaderLabels  setViewportMargins  sharedPainter  showEvent  sizeHintForColumn  sizeHintForRow  sortItems  startDrag  state  supportedDropActions  tabletEvent  takeHorizontalHeaderItem  takeItem  takeVerticalHeaderItem  timerEvent  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalHeaderItem  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualColumn  visualItemRect  visualRegionForSelection  visualRow  wheelEvent |

【详细描述】QTableWidget是QTableView的子类，主要的区别是QTableView可以使用自定义的数据模型来显示内容(也就是先要通过setModel来绑定数据源)，而QTableWidget则只能使用标准的数据模型，并且其单元格数据是QTableWidgetItem的对象来实现的(也就是不需要数据源，将逐个单元格内的信息填好即可)。这主要体现在QTableView类中有setModel成员函数，而到了QTableWidget类中，该成员函数变成了私有。使用QTableWidget就离不开QTableWidgetItem。QTableWidgetItem用来表示表格中的一个单元格，整个表格都需要用逐个单元格构建起来。

setItem：在QTableWidget表格控件的每个选项的单元空间里添加控件，参数为int，行；int, 列；str, 控件名，通常为QTableWidgetItem；

setCellWidget：在单元格里加入控件，参数为int，行；int, 列；str, 控件名，如QCombox等；

setHorizontalHeaderItem/ setverticalHeaderItem：设置QTableWidget表格控件的水平/垂直标签，参数为int, 表示表头中的位置；QTableWidgetItem，表头的内容；

setHorizontalHeaderLabels/ setVerticalHeaderLabels：设置QTableWidget表格控件的水平/垂直标签，参数为表头内容列表；

setEditTriggers：将表格变为禁止编辑，参数为枚举值，QAbstractItemView.EditTrigger对象【NoEditTriggers，不能对表格内容进行修改；CurrentChanged，任何时候都能对单元格修改；SelectedClicked，单击已选中的内容；AnyKeyPressed，按下任意键就能修改；AllEditTriggers，以上条件全包括】；

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

setTextAlignment：设置单元格内文字的对齐方式，参数为枚举值，Qt.AlignmentFlag对象；

setRowCount：设置QTableWidget表格控件的行数，参数为int；RowCoun：获取表格控件的行数；setColumnCount：设置QTableWidget表格控件的列数，参数为int；ColumnCount：获取表格控件的列数；

findItems：遍历表格查找对应的具体单元格，参数为查找对象str, 匹配方式为枚举值，Qt.MatchFlag对象，返回列表部件；

sortItems：将表格中的数据排序，参数为int, 行；第二参数为枚举值，Qt.SortOrder对象；

clearContents：清除表格工作区的所有内容，但是不清楚表头；clear：清除表格全部内容，包括表头；

currentColumn/currentRow：获取当前单元格的列号和行号；

insertColumn/insertRow：在列号/行号为Column/Row的列/行前面插入一列/行，如果Column/Row等于或大于总行数，则在表格最后添加一行，参数为int，如果为空，表示插入一个空行，但不会为单元格创建QTableWidgetItem对象，需要手动添加；

currentCellChanged：当单元格发生切换时触发该信号，并传递参数currentRow，currenColumn，previousRow，previousColumn；

currentItemChanged：当单元格发生切换时触发该信号，；并传递参数QTableWidgetItem；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTableWidgetItem

【继承关系】QtWidgets. QTableWidgetItem

|  |  |  |  |
| --- | --- | --- | --- |
| background  checkState  clone  column  data  flags  font  foreground  icon | isSelected  read  row  setBackground  setCheckState  setData  setFlags  setFont  setForeground | setIcon  setSelected  setSizeHint  setStatusTip  setText  setTextAlignment  setToolTip  setWhatsThis  sizeHint | statusTip  tableWidget  text  textAlignment  toolTip  type  whatsThis  write |

【详细描述】在QTableWiget表格中，每一个单元格是一个QTableWidgetItem对象，可以设置文本内容，字体，前景色，背景色，图标等，可以设置编辑和显示标志，每一个单元格还可以存储一个或多个自定义数据；

setTextAlignment：设置文字对齐方式，参数为枚举值，Qt.Alignment对象；

setBackground：设置单元格背景颜色，参数为Qbush对象；

setForeground：设置单元格前景色，即文字颜色，参数为Qbush对象；

setIcon：为单元格设置一个图标，参数为QIcon对象；

setFont：为单元格设置字体，参数为QFont对象；

setCheckState：设置单元格勾选状态，参数为Qt.CheckState对象；

setFlags：设置单元格的属性标志，参数为Qt.ItemFlag对象；

type：获得单元格的类型；

data：获得QTableWidgetItem的数据；

setData：设置QTableWidgetItem中的数据，参数1为枚举值，Qt.ItemDataRole对象；参数2为写入的内容；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QListView

【继承关系】QtWidgets.QListView –> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

【继承者】QHelpIndexWidget，QListWidget，QUndoView

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  batchSize  changeEvent  childEvent  clearPropertyFlags  closeEditor  closeEvent  commitData  connectNotify  contextMenuEvent  create  currentChanged  customEvent  dataChanged  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  edit  editorDestroyed  enterEvent  event  eventFilter  executeDelayedItemsLayout  flow  focusInEvent  focusNextChild | focusNextPrevChild  focusOutEvent  focusPreviousChild  gridSize  hideEvent  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indexAt  indexesMoved  initPainter  initStyleOption  inputMethodEvent  isIndexHidden  isRowHidden  isSelectionRectVisible  isSignalConnected  isWrapping  itemAlignment  keyPressEvent  keyReleaseEvent  layoutMode  leaveEvent  metric  modelColumn  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  movement | nativeEvent  paintEvent  receivers  rectForIndex  reset  resizeEvent  resizeMode  rowsAboutToBeRemoved  rowsInserted  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollTo  selectedIndexes  selectionChanged  selectionCommand  sender  senderSignalIndex  setBatchSize  setDirtyRegion  setFlow  setGridSize  setItemAlignment  setLayoutMode  setModelColumn  setMovement  setPositionForIndex  setResizeMode  setRootIndex  setRowHidden  setSelection  setSelectionRectVisible | setSpacing  setState  setUniformItemSizes  setViewMode  setViewportMargins  setWordWrap  setWrapping  sharedPainter  showEvent  spacing  startDrag  state  tabletEvent  timerEvent  uniformItemSizes  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewMode  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualRect  visualRegionForSelection  wheelEvent  wordWrap |

【详细描述】用于显示单列的列表数据，适用于以为数据的操作；

setModel：用于设置View所关联的Model；

selectedItem：选中Model中的条目；isSelected：判断Model中的某条目是否被选中

## QListWidget

【继承关系】QtWidgets.QListWidget -> QtWidgets.QListView –> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addItem  addItems  changeEvent  childEvent  clear  closeEditor  closeEvent  closePersistentEditor  commitData  connectNotify  contextMenuEvent  count  create  currentChanged  currentItem  currentItemChanged  currentRow  currentRowChanged  currentTextChanged  customEvent  dataChanged  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  dropIndicatorPosition  dropMimeData  edit  editItem  editorDestroyed | enterEvent  event  eventFilter  executeDelayedItemsLayout  findItems  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indexFromItem  initPainter  initStyleOption  inputMethodEvent  insertItem  insertItems  isIndexHidden  isPersistentEditorOpen  isSignalConnected  isSortingEnabled  item  itemActivated  itemAt  itemChanged  itemClicked  itemDoubleClicked  itemEntered  itemFromIndex  itemPressed  items  itemSelectionChanged | itemWidget  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mimeData  mimeTypes  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  openPersistentEditor  paintEvent  receivers  rectForIndex  removeItemWidget  resizeEvent  row  rowsAboutToBeRemoved  rowsInserted  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollToItem  selectedIndexes  selectedItems  selectionChanged  selectionCommand  sender  senderSignalIndex  setCurrentItem  setCurrentRow | setDirtyRegion  setItemWidget  setModel  setPositionForIndex  setSelection  setSelectionModel  setSortingEnabled  setState  setViewportMargins  sharedPainter  showEvent  sortItems  startDrag  state  supportedDropActions  tabletEvent  takeItem  timerEvent  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualItemRect  visualRegionForSelection  wheelEvent |

【详细描述】

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

## QTreeView

【继承关系】QtWidgets. QTreeView -> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

【继承者】QHelpContentWidget, QTreeWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  allColumnsShowFocus  autoExpandDelay  changeEvent  childEvent  closeEditor  closeEvent  collapse  collapseAll  collapsed  columnAt  columnCountChanged  columnMoved  columnResized  columnViewportPosition  columnWidth  commitData  connectNotify  contextMenuEvent  create  currentChanged  customEvent  dataChanged  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawBranches  drawFrame  drawRow  drawTree  dropEvent  dropIndicatorPosition  edit  editorDestroyed  enterEvent  event  eventFilter  executeDelayedItemsLayout | expand  expandAll  expanded  expandRecursively  expandsOnDoubleClick  expandToDepth  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  header  hideColumn  hideEvent  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indentation  indexAbove  indexAt  indexBelow  indexRowSizeHint  initPainter  initStyleOption  inputMethodEvent  isAnimated  isColumnHidden  isExpanded  isFirstColumnSpanned  isHeaderHidden  isIndexHidden  isRowHidden  isSignalConnected  isSortingEnabled  itemsExpandable  keyboardSearch  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mouseDoubleClickEvent | mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  paintEvent  receivers  reexpand  reset  resetIndentation  resizeColumnToContents  resizeEvent  rootIsDecorated  rowHeight  rowsAboutToBeRemoved  rowsInserted  rowsRemoved  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollTo  selectAll  selectedIndexes  selectionChanged  selectionCommand  sender  senderSignalIndex  setAllColumnsShowFocus  setAnimated  setAutoExpandDelay  setColumnHidden  setColumnWidth  setDirtyRegion  setExpanded  setExpandsOnDoubleClick  setFirstColumnSpanned  setHeader  setHeaderHidden  setIndentation  setItemsExpandable | setModel  setRootIndex  setRootIsDecorated  setRowHidden  setSelection  setSelectionModel  setSortingEnabled  setState  setTreePosition  setUniformRowHeights  setViewportMargins  setWordWrap  sharedPainter  showColumn  showEvent  sizeHintForColumn  sortByColumn  startDrag  state  tabletEvent  timerEvent  treePosition  uniformRowHeights  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualRect  visualRegionForSelection  wheelEvent  wordWrap |

【详细描述】用于显示树状结构的数据，适用于树状结构数据的操作；

setColumnWidth：将指定列的宽度设置为指定的值，参数 1. int，指定的列；2. int，指定列的宽度；

expandAll：展开所有的树形节点；

invisibleRootItem：返回树形控件中不可见的根选项（Root Item）；

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

## QTreeWidget

【继承关系】QtWidgets.QTreeWidget -> QTreeWidget.QTreeView -> QtWidgets. QAbstractItemView -> QtWidgets. QAbstractScrollArea -> QtWidgets. QFrame -> QtWigets. QtWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addTopLevelItem  addTopLevelItems  changeEvent  childEvent  clear  closeEditor  closeEvent  closePersistentEditor  collapseItem  columnCount  columnCountChanged  columnMoved  columnResized  commitData  connectNotify  contextMenuEvent  create  currentChanged  currentColumn  currentItem  currentItemChanged  customEvent  destroy  dirtyRegionOffset  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawBranches  drawFrame  drawRow  drawTree  dropEvent  dropIndicatorPosition  dropMimeData  edit  editItem  editorDestroyed | enterEvent  event  eventFilter  executeDelayedItemsLayout  expandItem  findItems  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  headerItem  hideEvent  horizontalOffset  horizontalScrollbarAction  horizontalScrollbarValueChanged  indexFromItem  indexOfTopLevelItem  indexRowSizeHint  initPainter  initStyleOption  inputMethodEvent  insertTopLevelItem  insertTopLevelItems  invisibleRootItem  isFirstItemColumnSpanned  isIndexHidden  isPersistentEditorOpen  isSignalConnected  itemAbove  itemActivated  itemAt  itemBelow  itemChanged  itemClicked  itemCollapsed  itemDoubleClicked  itemEntered  itemExpanded | itemFromIndex  itemPressed  itemSelectionChanged  itemWidget  keyPressEvent  keyReleaseEvent  leaveEvent  metric  mimeData  mimeTypes  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveCursor  moveEvent  nativeEvent  openPersistentEditor  paintEvent  receivers  reexpand  removeItemWidget  resizeEvent  rowHeight  rowsAboutToBeRemoved  rowsInserted  rowsRemoved  scheduleDelayedItemsLayout  scrollContentsBy  scrollDirtyRegion  scrollToItem  selectedIndexes  selectedItems  selectionChanged  selectionCommand  sender  senderSignalIndex  setColumnCount  setCurrentItem | setDirtyRegion  setFirstItemColumnSpanned  setHeaderItem  setHeaderLabel  setHeaderLabels  setItemWidget  setModel  setSelection  setSelectionModel  setState  setViewportMargins  sharedPainter  showEvent  sizeHintForColumn  sortColumn  sortItems  startDrag  state  supportedDropActions  tabletEvent  takeTopLevelItem  timerEvent  topLevelItem  topLevelItemCount  updateEditorData  updateEditorGeometries  updateGeometries  updateMicroFocus  verticalOffset  verticalScrollbarAction  verticalScrollbarValueChanged  viewOptions  viewportEvent  viewportMargins  viewportSizeHint  visualItemRect  visualRegionForSelection  wheelEvent |

【详细描述】

insertTopLevelItems：在视图的顶层索引中插入项目列表；

selectedItems：返回所有选项的非隐藏项目的列表；

setSelectionModel：设置单个选中和多个选中，参数为枚举值，QAbstractItemView.SelectionModel对象【NoSelection，不能选择；SingleSelection，选中单个目标；MultiSelection，选中多个目标】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QTreeWidgetItem

【继承关系】QtWidgets. QTreeWidgetItem

|  |  |  |  |
| --- | --- | --- | --- |
| addChild  addChildren  background  checkState  child  childCount  childIndicatorPolicy  clone  columnCount  data  emitDataChanged  flags  font  foreground  icon | indexOfChild  insertChild  insertChildren  isDisabled  isExpanded  isFirstColumnSpanned  isHidden  isSelected  parent  read  removeChild  setBackground  setCheckState  setChildIndicatorPolicy  setData | setDisabled  setExpanded  setFirstColumnSpanned  setFlags  setFont  setForeground  setHidden  setIcon  setSelected  setSizeHint  setStatusTip  setText  setTextAlignment  setToolTip  setWhatsThis | sizeHint  sortChildren  statusTip  takeChild  takeChildren  text  textAlignment  toolTip  treeWidget  type  whatsThis  write |

【详细描述】树形结构是通过QTreeWidget和QTreeWidgetItem类实现的，其中QTreeWidgetItem类实现了节点的添加；‘

setText

## QColumnView

【详细描述】多用于QListView显示树状层次结构，树状结构的一层用一个QListView显示；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QModelIndex

【继承关系】QtCore. QModelIndex

|  |  |  |  |
| --- | --- | --- | --- |
| child  column  data  flags | internalId  internalPointer  isValid  model | parent  row  sibling  siblingAtColumn | siblingAtRow |

【详细描述】模型索引，用于定位数据模型中的数据，初始化方法1，QAbstractItemModel.index(row->int, column->int, [parent=QModelIndex() ] )，通过给定row , column, parent 的index返回模型中的项索引；方法2，QModelIndex.child (row=int, column=int)，返回存储在指定row和column中的模型索引的子元素。

row：返回模型索引所在的列；column：返回模型索引所在的列；

data：返回模型索引指定的role值，默认为DisplayRole；

flags：返回索引指向单元格的标志，返回Qt.ItemFlags对象；

isValid：判断索引是否有效，如查出索引的范围则返回False；

model：返回该索引所属的QAbstractItemModel模型对象；

sibling：返回指定行列的兄弟索引，参数row-> int，column-> int；siblingAtColumn：返回与当前索引同行，不同列的索引，参数column-> int；siblingAtRow：当前与当前索引同列，不同行的索引，参数row-> int；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractItemModel

【继承关系】QtCore.QAbstractItemModel -> QtCore.QObject

【继承者】QAbstractListModel, QAbstractProxyModel, QAbstractTableModel, QConcatenateTablesProxyModel, QIdentityProxyModel, QSortFilterProxyModel, QStringListModel, QTransposeProxyModel, QStandardItemModel, QHelpContentModel, QHelpIndexModel, QSqlQueryModel, QSqlRelationalTableModel, QSqlTableModel, QDirModel, QfileSystemModel

|  |  |  |  |
| --- | --- | --- | --- |
| beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  buddy  canDropMimeData  canFetchMore  changePersistentIndex  changePersistentIndexList  checkIndex  columnCount  columnsAboutToBeInserted  columnsAboutToBeMoved  columnsAboutToBeRemoved  columnsInserted  columnsMoved  columnsRemoved | createIndex  data  decodeData  dropMimeData  encodeData  endInsertColumns  endInsertRows  endMoveColumns  endMoveRows  endRemoveColumns  endRemoveRows  endResetModel  fetchMore  flags  hasChildren  hasIndex  headerData  headerDataChanged  index  insertColumn | insertColumns  insertRow  insertRows  itemData  layoutAboutToBeChanged  layoutChanged  match  mimeData  mimeTypes  modelAboutToBeReset  modelReset  moveColumn  moveColumns  moveRow  moveRows  parent  parent  persistentIndexList  removeColumn  removeColumns | removeRow  removeRows  resetInternalData  revert, submit  roleNames  rowCount  rowsAboutToBeInserted  rowsAboutToBeMoved  rowsAboutToBeRemoved  rowsInserted  rowsMoved  rowsRemoved  setData  setHeaderData  setItemData  sibling  sort  span  supportedDragActions  supportedDropActions |

【详细描述】

模型/视图编程介绍

在model/view构架中，QAbstractItemModel类为view（视图）和delegates（代理）使用数据提供了标准接口。不管数据在底层以何种数据结构存储，QAabstractItemModel的子类会以层次结构的形式来表示数据，结构中包含了数据项表。我们按这种约定来访问model中的数据项，但这个约定不会对如何显示这些数据有任何限制。数据发生改变时，model通过信号/槽机制来通知关联的views。

模型:

所有的模型都基于QAbstractItemModel类。这个类定义了一个使用视图和委托来访问数据的接口。数据本身不是必须要存储在模型中，可以在一个数据结构或一个单独的类、文件、数据库、或其它一些应用组件。

QAbstractItemModel为数据提供了一个接口，使其能够在model/view架构中与其他组件进行交互。但它不能直接实例化，但能子类化它来创建新的model。它足够的灵活性来处理表格、列表、树形式的数据视图。然而，实现新的列表和类似于表的数据结构模型时，QAbstractListModel和QAbstractTableModel类是更好的起点，因为它们提供了适当的常用的功能的默认实现。这些类可以派生子类，用来提供支持特定种类的列表和表格的模型。

如果这些标准模型不能满足要求，则可以继承化QAbstractItemModel、QAbstractListModel或QAbstractTableModel来创建自定义模型。

视图:

为实现各种不同的目的PYQT5提供了完整的视图方案： QListView显示项目列表，QTableView中从一个表模型中显示数据，QTreeView则显示了层次列表的数据模型项目。每个类是基于QAbstractItemView抽象基类。虽然这些类是准备使用的实现，他们也可以被子类化，以提供自定义的视图。

QListWidget，QTreeWidget和QTableWidget比视图类（QListView，QTreeView和QTableView）灵活性差，且不能与任意的模型使用。

代理：在视图与模型之间交互操作时提供临时编辑组件的功能。模型向视图提供数据是单向的，当需要在视图上编辑数据时，代理功能会为编辑数据提供一个编辑器，这个编辑器获取模型的数据、接受用户编辑的数据后有提交给模型。

QAbstractItemDelegate在模型/视图框架中代表抽象的基类。默认的委托实现由QStyledItemDelegate提供，这被Qt的标准视图用作默认的委托。然而，QStyledItemDelegate和QItemDelegate独立替代绘画，且为视图项提供编辑器。它们之间的区别在于QStyledItemDelegate使用当前样式来绘制项目。因此，建议实现自定义委托或当与Qt样式表一起使用时，使用QStyledItemDelegate作为基类。

排序:

在模型/视图结构中有两种排序方式，选择哪种方式取决于你的基础模型。

方法一，如果你的模型是可排序的，也就是说，如果重新实现了QabstractItemModel.sort()方法，QTableView和QTreeView都提供了一个API，允许以编程方式排序来排序模型数据。此外，可以启用交互式排序（即允许用户将数据通过单击视图的标题进行排序），QheaderView.sortIndicatorChanged()信号分别连接到QTableView.sortByColumn()槽或QtreeView.sortByColumn()槽。

方法二，如果模型没有所需的接口，或者如果想使用一个列表视图来显示数据，使用代理模型呈现数据视图之前应转换模型的结构，如QSortFilterProxyModel()。

insertRow：在数据模型的row行前面插入一个记录，如果row大于或者等于数据模型的总行数，则在最后一行添加记录Item，参数1. row[int]行；2. Item[QModelIndex对象]；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractListModel

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStringListModel

【继承关系】QtCore.QStringListModel -> QtCore.QAbstractListModel -> QtCore.QAbstractItemModel -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| data  flags  insertRows  itemData | moveRows  removeRows  rowCount  setData | setItemData  setStringList  sibling  sort | stringList  supportedDropActions |

【详细描述】用于处理字符串列表数据的数据模型类，通常与QListView组合成Model/View结构；

setStringList：初始化数据模型的字符串列表内容，参数为list；stringList：返回数据模型内的字符串列表；

insertRows：插入行，其中参数row->int是行号，表示在row行之前插入一行；如不带参数，表示在列表尾部添加一个空行；

setData：将新增的项添加到模型的index位置，参数1. Index->QModelIndex；2. str，添加的内容；3. Qt.ItemDataRole对象，用于表示添加的属性；

rowCount：返回列表的总行数；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractProxyModel

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSortFilterProxyModel

【详细描述】与其他数据模型结合，提供排序和过滤功能的数据模型类

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractTableModel

【继承关系】QtCore.QAbstractTableModel -> QtCore.QAbstractItemModel -> QtCore.QObject

【继承者】QSqlQueryModel, QSqlRelationalTableModel, QSqlTableModel

|  |  |  |  |
| --- | --- | --- | --- |
| sibling  parent | flags | dropMimeData | index |

【详细描述】QAbstractTableModel提供了一个抽象的模型，能够被子类化创建table模型；QAbstractTableModel为model提供了一个标准的接口，能够以一个两维数组项表示他们的数据。不能直接使用，必须被子类化。因为比QAbstractItemModel提供了更多具体的接口，并不适合tree视图的使用，尽管它也用于给QListView提供数据。如果你需要表示一个简单的关于items的列表，并仅需要一个包含单列数据的模型，子类化QAbstractListModel可能更合适。

【常用函数】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlDatabase

【继承关系】QtSql.QSqlDatabase

|  |  |  |  |
| --- | --- | --- | --- |
| addDatabase  cloneDatabase  close  commit  connectionName  connectionNames  connectOptions  contains  database  databaseName | driver  driverName  drivers  exec  exec\_  hostName  isDriverAvailable  isOpen  isOpenError  isValid | lastError  numericalPrecisionPolicy  open  password  port  primaryIndex  record  registerSqlDriver  removeDatabase  rollback | setConnectOptions  setDatabaseName  setHostName  setNumericalPrecisionPolicy  setPassword  setPort  setUserName  tables  transaction  userName |

【详细描述】用于建立于数据库的连接；

drivers：显示支持的数据库驱动；

addDatabase：设置连接数据库的数据库驱动类型，返回QSqlDatabase类型的对象，即创建的数据库连接，参数1. 为dbType数据库驱动，常用数据库驱动[1. QMYSQL，Mysql驱动程序；2. QSQLITE，SQLite3驱动程序]；2. ConnectionName，将数据库的连接命名（可选）；

setDatabaseName：设置所连接的数据库名称dbName，如果数据库不存在，则会自动创建；如果存在，则采用该数据库；

setHostName：设置安装数据库的主机名称；

setUserName：设置数据库用户名；

setPassword：设置连接对象的密码（如果有）；

connectionName：获取数据库连接ConnectionName的名称;

removeDatabase：删除数据库连接，参数为ConnectionName；

contains：查看数据库连接ConnectionName的连接是否存在，参数1. ConnectionName, 表示连接的名称（可选）， 其中qt\_sql\_default\_connection表示Qt默认连接名称；

database：获取名称为ConnectionName的数据库连接；

commit：提交事务，如果执行；rollback：回滚数据库事务；

open：打开数据库连接，返回bool值，表示数据库是否成功打开；close：关闭数据库连接；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlDriver

【详细描述】用于构建数据库驱动；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlError

【详细描述】SQL数据库出错消息，可以用于访问上一次出错的信息；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlQuery

【继承关系】QtSql.QSqlQuery

|  |  |  |  |
| --- | --- | --- | --- |
| addBindValue  at  bindValue  boundValue  boundValues  clear  driver  exec  exec\_ | execBatch  executedQuery  finish  first  isActive  isForwardOnly  isNull  isSelect  isValid | last  lastError  lastInsertId  lastQuery  next  nextResult  numericalPrecisionPolicy  numRowsAffected  prepare | previous  record  result  seek  setForwardOnly  setNumericalPrecisionPolicy  size  value |

【详细描述】执行各种SQL语句的类；

prepare：设置准备的SQL语句，一般用于带参数的SQL语句，参数采用““”：参数名“的形式；

bindValue：设置SQL语句中参数placeholder的值为val，参数1. placeholder；参数2. val；

ecec：执行有parpare和bindValue设置的带参数的SQL语句，或者直接执行一个不带参数的SQL语句；

executedQuery：返回最后一次查询的字符串；

isActive：如果成功执行了exce函数，则返回True；

isSelect：如果执行的SQL语句是SELECT语句，则返回True；

record：返回一条记录，当没有参数时返回一条空记录，只有字段名的QSqlRecord对象；当参数为row，行号时，返回含有字段定义和数据的QSqlRecord对象；

value：返回当前记录中名为filedname的字段的值，参数为filedname[QString]；

isNull：判断一个字段是否为空，当Query非活动，未定位在有效记录，无此字段或字段为空时，返回True；

size：对于SELECT语句，返回查询的记录条数，其他语句返回-1；

numRowsAffected：返回SQL语句影响的记录条数，如DELETE语句删除记录的条数，对于SELECT语句无定义；

first/previous/next/last：在记录中移动，isActive和isSelect都为True时才有效；

seek：定位到指定序号为Index的记录；

at：返回当前记录的序号；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlRecord

【继承关系】QtSql.QSqlRecord

|  |  |  |  |
| --- | --- | --- | --- |
| append  clear  clearValues  contains  count | field  fieldName  indexOf  insert  isEmpty | isGenerated  isNull  keyValues  remove  replace | setGenerated  setNull  setValue  value |

【详细描述】记录了数据表的字段信息和一条记录的数据内容；

value：返回字段名称为name或索引为Index的字段的值，参数QString或int，返回QVariant对象；setValue：设置字段名称为name的字段的值为Val，参数1. name, 2. Val；

field：返回指定索引的字段，或者根据名称得到字段，返回QSqlField对象；

IndexOf：返回字段名称为name的字段的序号，参数为Qstring,如果字段不存在，则返回-1；

count：返回记录的字段数量int；

fieldName：根据索引获得相应的字段的名称，参数为Int，返回QString对象；

append：追加字段，参数为QSqlField类型；

clear：清理记录的所有字段定义和数据；

clearValues：清除所有字段的数据，将字段数据内容设置为None；

contains：判断记录是否包含名称为name的字段，参数为Qstring, 返回bool类型；

isEmpty：若记录中没有字段则返回True，否则返回False；

isNull：判断名字为name的字段数据是否为空，参数name, Qstring，返回bool类型；setNull：设置名称为name的字段的值为空，参数name，Qstring；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlField

【继承关系】QtSql.QSqlField

|  |  |  |  |
| --- | --- | --- | --- |
| clear  defaultValue  isAutoValue  isGenerated  isNull  isReadOnly  isValid | length  name  precision  requiredStatus  setAutoValue  setDefaultValue  setGenerated | setLength  setName  setPrecision  setReadOnly  setRequired  setRequiredStatus  setSqlType | setTableName  setType  setValue  tableName  type  typeID  value |

【详细描述】用于定义字段信息和数据的类;

name：返回字段的名称；

value：返回字段的值，返回值是QVariant类型；setValue：设置字段的值为Value；

type：返回字段的类型，QMetaType.Type对象；

length：返回字段的长度。如果返回值是负数，意味着信息无法从数据库获得。

precision：返回该字段的精度；这只对数值类型有意义，如果返回负数，意味着信息无法从数据库获得；

clear：清除字段数据，设置为空。如果字段为只读的，则不清除；

isNull：判断字段值是否为空；

setReadOnly：设置一个字段是否为只读，参数为bool类型，只读的字段不能用setValue函数设置值，也不能用Clear函数清除值；

defaultValue：返回字段的默认值（可能为空（NULL））。

isAutoValue：如果值是由数据库自动生成的返回true，例如，自增的主键值。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlIndex

【继承关系】QtSql.QSqlIndex –> QtSql.QSqlRecord

|  |  |  |  |
| --- | --- | --- | --- |
| append  cursorName | isDescending  name | setCursorName  setDescending | setName |

【详细描述】操作数据库的索引的类

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlQueryModel

【继承关系】QtCore.QSqlQueryModel -> QtCore.QAbstractTableModel -> QtCore.QAbstractItemModel -> QtCore.QObject

【继承者】QSqlRelationalTableModel, QSqlTableModel

|  |  |  |  |
| --- | --- | --- | --- |
| beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  canFetchMore  changePersistentIndex  changePersistentIndexList  childEvent  clear | columnCount  connectNotify  createIndex  customEvent  data  decodeData  disconnectNotify  encodeData  endInsertColumns  endInsertRows  endMoveColumns  endMoveRows | endRemoveColumns  endRemoveRows  endResetModel  fetchMore  headerData  indexInQuery  insertColumns  isSignalConnected  lastError  persistentIndexList  query  queryChange | receivers  record  removeColumns  resetInternalData  roleNames  rowCount  sender  senderSignalIndex  setHeaderData  setLastError  setQuery  timerEvent |

【详细描述】SQL查询结果数据的制度数据模型类，用于SELECT查询结果数据记录的只读显示，不可以修改数据；其于QSqlTableModel存在一下区别：1. QSqlTableModel只能设置一个数据表，查询出数据表的所有字段的数据，而QSqlQueryModel通过SELECT语句，可以从一个或者多个数据表里灵活查询所需要的数据；2. QSqlTableModel得到的数据是可以编辑修改的，而QSqlQueryModel查询得到的数据是只读的；3. QSqlTableModel和QSqlQueryModel都可以作为QTableView和QDataWidgetMapper的数据模型；

【常用函数】

setQuery：设置SELECT语句从一个或多个数据表中查询数据记录，注，只能用SELECT语句；query：获得查询的QSqlQuery对象；

setHeaderData：设置每个字段的表头数据，参数1. int, 字段的序号；2. Qt.Orientation对象，表示方向；3. 表头名称；

lastError：用于获取错误的具体信息；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlTableModel

【继承关系】QtCore.QSqlTableModel –> QtCore.QSqlQueryModel -> QtCore.QAbstractTableModel -> QtCore.QAbstractItemModel -> QtCore.Qobject

|  |  |  |  |
| --- | --- | --- | --- |
| beforeDelete  beforeInsert  beforeUpdate  beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  changePersistentIndex  changePersistentIndexList  childEvent  clear  connectNotify  createIndex  customEvent  data  database | decodeData  deleteRowFromTable  disconnectNotify  editStrategy  encodeData  endInsertColumns  endInsertRows  endMoveColumns  endMoveRows  endRemoveColumns  endRemoveRows  endResetModel  fieldIndex  filter  flags  headerData  indexInQuery  insertRecord  insertRowIntoTable | insertRows  isDirty  isSignalConnected  orderByClause  persistentIndexList  primaryKey  primaryValues  primeInsert  queryChange  receivers  record  removeColumns  removeRows  resetInternalData  revert  revertAll  revertRow  rowCount  select | selectRow  selectStatement  sender  senderSignalIndex  setData  setEditStrategy  setFilter  setLastError  setPrimaryKey  setQuery  setRecord  setSort  setTable  sort  submit  submitAll  tableName  timerEvent  updateRowInTable |

【详细描述】编辑一个单一数据表的模型数据类型，可以获取数据表的全部记录，其结果是可编辑的；初始化QSqlTableModel(QSqlDatabase)，需要传递一个QSqlDatabase对象作为连接的数据库对象，如果不传递数据库对象，会自动使用应程序里已经创建的数据库对象作为连接的数据库对象；通常与 QTableView 组合成 Model/View 结构；

setTable：设置需要连接的数据表；

setEditStrategy：设置编辑策略，参数为枚举值，QSqlTableModel.EditStrategy[1. OnFieldChange，所有变更实时更新到数据库中；2. OnRowChange，当用户选择不同行时，在当前行进行变更；3. OnManualSubmit，手动提交，不自动提交]

fieldIndex：返回字段fieldName的索引，如果没有相应字段返回-1，参数为QString；

select：选取整个表的所有行

setSort：排序操作，参数1. id属性即第i列；2. 升序排列，枚举值，Qt.SortOrder对象；

record：返回一条记录，当没有参数时返回一条空记录，只有字段名的QSqlRecord对象；当参数为row，行号时，返回含有字段定义和数据的QSqlRecord对象；

isDirty：用于判断是否有未提交的修改，返回bool对象；

submitAll：用于将数据表中所有为提交的修改保存到数据库中；

revertAll：取消所有的修改；

setRecord：修改某一行记录，参数1. row[int]；2. QSqlRecord对象，同时需要注意在需要时需要调用submitAll函数；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlRelation

【继承关系】QtSql.QSqlRelation

|  |  |  |  |
| --- | --- | --- | --- |
| displayColumn  indexColumn | isValid  swap | tableName |  |

【详细描述】用于存储SQL外键信息的类，用于QSqlRelationTableModel数据源中设置代码字段与关联数据表的关系，初始化QSqlRelation(aTableName，indexCol，displayCol)；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlRelationalTableModel

【继承关系】QtSql.QSqlRelationalTableModel -> QtSql.QSqlTableModel -> QtSql.QSqlQueryModel -> QtCore.QAbstractTableModel -> QtCore.QAbstractItemModel -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  changePersistentIndex  changePersistentIndexList  childEvent  clear  connectNotify  createIndex | customEvent  data  decodeData  deleteRowFromTable  disconnectNotify  encodeData  endInsertColumns  endInsertRows  endMoveColumns  endMoveRows  endRemoveColumns  endRemoveRows  endResetModel | indexInQuery  insertRowIntoTable  isSignalConnected  orderByClause  persistentIndexList  primaryValues  queryChange  receivers  relation  relationModel  removeColumns  resetInternalData  revertRow | select  selectStatement  sender  senderSignalIndex  setData  setJoinMode  setLastError  setPrimaryKey  setQuery  setRelation  setTable  timerEvent  updateRowInTable |

【详细描述】用于一个数据表的可编辑的数据模型，可以通过关系将代码字段与代码便关联，将代码表的编辑转换为直观的列表选择，初始化；

setRelation：设置代码字段的关系数据表和关联字段，参数1，column，主表中代码字段的序号；2. relation，QSqlRelation类型，用于表示数据表的关系；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSqlRelationalDelegate

【继承关系】QtSql.QSqlRelationalDelegate -> QtWidgets.QItemDelegate –> QtWidgets.QAbstractItemDelegate –> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  connectNotify  createEditor  customEvent  disconnectNotify | drawBackground  drawCheck  drawDecoration  drawDisplay  drawFocus | editorEvent  eventFilter  isSignalConnected  receivers  sender | senderSignalIndex  setEditorData  setModelData  timerEvent |

【详细描述】用于QSqlRelationalTableModel的一个代码字段的显示和编辑的代理组件，一般是一个QComboBox组件，下拉列表中自动填充代码表的代码字段对应的实际内容；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDataWidgetMapper

【继承关系】QtWidgets.QDataWidgetMapper -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| addMapping  childEvent  clearMapping  connectNotify  currentIndex  currentIndexChanged  customEvent  disconnectNotify  isSignalConnected | itemDelegate  mappedPropertyName  mappedSection  mappedWidgetAt  model  orientation  receivers  removeMapping  revert | rootIndex  sender  senderSignalIndex  setCurrentIndex  setCurrentModelIndex  setItemDelegate  setModel  setOrientation  setRootIndex | setSubmitPolicy  submit  submitPolicy  timerEvent  toFirst  toLast  toNext  toPrevious |

【详细描述】用于建立界面组件于数据模型之间的映射，可以将界面上的QLineEdit，QComboBox等组件与数据模型的一个字段关联起来，实现字段内容的自动显示；

setModel：设置关联数据模型；

setSubmitPolicy：设置数据提交策略，参数为枚举值，QDataWidgetMapper.SubmitPolicy对象[1. AutoSubmit，自动提交，若提交策略为自动时，行切换时将自东提交修改；2. ManualSubmit，手动提交]

addMapping：设置界面组件与数据模型的字段映射，参数1. widget，指某个界面组件；2. section，关联的字段序号；

toFirst/toPrevious/toNext/toLast：用于在记录间移动；

setCurrentIndex：根据映射的行号更新数据，参数为int，表示行号；

setCurrentModelIndex：根据映射的行号更新数据，参数为QModelIndex，表示索引；

revert/submit：用于手动取消/提交当前记录的修改；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStandardItemModel

【继承关系】QtGui.QStandardItemModel –> QtCore.QAbstractItemModel -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| appendColumn  appendRow  beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  changePersistentIndex  changePersistentIndexList  childEvent  clear  clearItemData  columnCount  connectNotify  createIndex  customEvent  data  decodeData  disconnectNotify | dropMimeData  encodeData  endInsertColumns  endInsertRows  endMoveColumns  endMoveRows  endRemoveColumns  endRemoveRows  endResetModel  findItems  flags  hasChildren  headerData  horizontalHeaderItem  index  indexFromItem  insertColumn  insertColumns  insertRow  insertRows  invisibleRootItem | isSignalConnected  item  itemChanged  itemData  itemFromIndex  itemPrototype  mimeData  mimeTypes  parent  persistentIndexList  receivers  removeColumns  removeRows  resetInternalData  rowCount  sender  senderSignalIndex  setColumnCount  setData  setHeaderData  setHorizontalHeaderItem | setHorizontalHeaderLabels  setItem  setItemData  setItemPrototype  setItemRoleNames  setRowCount  setSortRole  setVerticalHeaderItem  setVerticalHeaderLabels  sibling  sort  sortRole  supportedDropActions  takeColumn  takeHorizontalHeaderItem  takeItem  takeRow  takeVerticalHeaderItem  timerEvent  verticalHeaderItem |

【详细描述】标准的基于数据项的数据模型类，用于记录表格数据，每个项数据可以是任何数据类型，通常与 QTableView 组合成 Model/View 结构；

data：返回index索引位置的数据，参数1. index, QModelIndex数据类型；2. Qt.ItemDataRole对象，表示获取的数据属性；

item：根据行号和列号返回QStandardItem指针，参数为row, int；column, int；

clear：清除模型中的数据；

clearItemData：清除QModelIndex中的数据项；

findItems：根据指定的column，指定的text，用指定的flags来查询出匹配的项（QStandardItem）。指定的flags默认为相等，可以进行正则匹配。

horizontalHeaderItem：返回指定列的行表头项，返回QStandardItem对象；setHorizontalHeaderItem：设置指定列的行表头，参数为1. in，column；2. QStandardItem；

VerticalHeaderItem：返回指定行的列表头项，返回QStandardItem对象；setVerticalHeaderItem：设置指定行的列表头，参数为1. in，column；2. QStandardItem；

setHorizontalHeaderLabels：设置水平表头，参数为QStringList对象；setVerticalHeaderLabels：设置列表头，参数为QStringList对象

indexFromItem：返回查询指定QStandardItem的QModelIndex；itemFromIndex：获取指定QModelIndex的项数据对象，返回QStandardItem；

setColumnCount：设置列数量，超出设置的列会丢弃，参数为int；setRowCount：设置行数量，超出设置的行会丢弃，，参数为int；

invisibleRootItem：返回模型最顶层的根节点（Root Item）,返回一个QStandardItem对象；

setItem：为模型的某个行列位置设置Item，参数为row[int]，column[int]，Item[QStandartItem]；

appendRow：向列表中添加一个列数据项；

insertRow：插入一行，参数row[int]，items[QStandartItem]类型的列表；

removeRow：删除QModelIndex所指定的行，参数为row[int]；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStandardItem

【继承关系】QtGui.QStandardItem

|  |  |  |  |
| --- | --- | --- | --- |
| accessibleDescription  accessibleText  appendColumn  appendRow  appendRows  background  checkState  child  clearData  clone  column  columnCount  data  emitDataChanged  flags  font  foreground  hasChildren  icon  index  insertColumn | insertColumns  insertRow  insertRows  isAutoTristate  isCheckable  isDragEnabled  isDropEnabled  isEditable  isEnabled  isSelectable  isTristate  isUserTristate  model  parent  read  removeColumn  removeColumns  removeRow  removeRows  row  rowCount | setAccessibleDescription  setAccessibleText  setAutoTristate  setBackground  setCheckable  setCheckState  setChild  setColumnCount  setData  setDragEnabled  setDropEnabled  setEditable  setEnabled  setFlags  setFont  setForeground  setIcon  setRowCount  setSelectable  setSizeHint  setStatusTip | setText  setTextAlignment  setToolTip  setTristate  setUserTristate  setWhatsThis  sizeHint  sortChildren  statusTip  takeChild  takeColumn  takeRow  text  textAlignment  toolTip  type  whatsThis  write |

【详细描述】QStandardItem是一个数据结构，他可以存储一个cell的各种信息，比如文本、图标、是否可选、字体、别景色、前景色等等。并且QStandardItem可以有孩子和兄弟，他是为model提供数据存储的节点，初始化item=QStandardItem(QString)

appendRow：向列表中添加一个数据项；appendColumn：纵向添加QStandardItem；

setData：为数据模型的一个数据项设置不同角色的数据，即将[role, value]对存入，其参数1. value，为任意类型的数据；[role，枚举值，Qt.ItemDataRole对象；]data：返回项的数据；

setBackground：设置背景刷子；background：返回背景刷子；setFont：设置文字字体； font：返回文字字体；setForeground：设置字体颜色；foreground：返回字体颜色；

setCheckable：在设置复选框；setCheckState：设置复选框状态；checkState：返回复选框状态；

setTextAlignment：设置文字对齐方式，参数为

data：返回角色role对应的数据子项，参数为role；

setFlags：设置数据项的一些标志（属性），参数为枚举值，Qt.ItemFlag对象；

setDragEnabled：设置QWidget对象A可以拖动，参数为Bool；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QItemSelectionModel

【继承关系】QtGui.QItemSelectionModel -> QtCore.Object

|  |  |  |  |
| --- | --- | --- | --- |
| clear  clearCurrentIndex  clearSelection  columnIntersectsSelection  currentChanged  currentColumnChanged  currentIndex | currentRowChanged  emitSelectionChanged  hasSelection  isColumnSelected  isRowSelected  isSelected  model | modelChanged  reset  rowIntersectsSelection  select  selectedColumns  selectedIndexes  selectedRows | selection  selectionChanged  setCurrentIndex  setModel |

【详细描述】选择模型，用于跟踪视图组件的单元格的选择情况，获取当前选择的行，列信息，并在选择的单元格发生变化时触发currentChanged信号，在当前行发生变化时，触发currentRowChanged信号，需要指定一个QStandardItemModel类数据模型，这样就与数据模型建立了关联，可以反映数据模型的项数据选择操作；

currentIndex：获取当前单元格的QModelIndex索引，当前单元格是键盘可以操作和获取鼠标焦点的单元格；setCurrentIndex：设置QModelIndex位置被选中，参数1，选中的位置QModelIndex对象；2. QItemSelectionModel. SelectionFlags对象

clearSelection：清除选择；

hasSelection：返回是否有单元被选择的判断，返回一个bool值；

isColumnSelected：

selectedIndexes：返回一个元素为QModelIndex类型的列表，包括所有被选中的单元格的模型索引；

select：利用Model左上角和右下角的坐标构建一个QItemSelection对象，参数1，selection为需要选择的选区；2，QItemSelectionModel. SelectionFlags对象，用于表示选区的标志位；

currentChanged：当选择的当前单元格发生变化时发射此信号，这个信号传递两个参数：第一个是新的当前项的QModelIndex，第二个是上一个当前项的QModelIndex。

QItemSelectionModel. SelectionFlags对象：1. NoUpdate，不执行任何选择操作；2. Clear，清除所有选择；3. Select，所有指定的索引都会被选择；4. Deselect，所有指定的索引都会去选择；5. Toggle，所有指定的索引都会切换当前的选择状态，选中的变为未选中，未选中的变为选中；6. Current，当前项更新为指定项，选择状态保持不变；7. Rows，所有指定索引项的选择将扩展到所有项所在行；8. Columns，所有指定索引项的选择将扩展到所有项所在列；9. SelectCurrent，等同于Select& Current；10. ToggleCurrent，等同于Toggle& Current；11. ClearAndSelect，等同于Clear& Select；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QAbstractItemDelegate

【继承关系】QtWidgets. QAbstractItemDelegate –> QtCore.QObject

【继承者】QItemDelegate, QStyledItemDelegate

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  closeEditor  commitData  connectNotify  createEditor  customEvent | destroyEditor  disconnectNotify  editorEvent  helpEvent  isSignalConnected  paint | receivers  sender  senderSignalIndex  setEditorData  setModelData  sizeHint | sizeHintChanged  timerEvent  updateEditorGeometry |

【详细描述】所有代理类的抽象基类；

createEditor：创建一个用于编辑模型数据的widget组件，并作为函数的返回值，参数1. parent，代理组件的父容器对象；2. option，QStyleOpitionViewItem类型变量，用于对创建的组件进行高级设置；3. index，QModelIndex变量，是关联数据项的模型索引；

setEditorData：从数据模型获取数据，供widget组件进行编辑，参数1. editor指向代理编辑器组件；2. index，关联的数据单元的所行索引；

setModelData：将widget上的数据更新到数据模型，参数1. editor，所创建的代理编辑器组件；2. model，关联的数据模型；3. Index, 关联的数据单元的所行索引；

updateEditorGeometry: 用于给widget组件设置合适的大小；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QItemDelegate

【继承关系】QtWidgets.QItemDelegate –> QtWidgets.QAbstractItemDelegate –> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  connectNotify  createEditor  customEvent  disconnectNotify  drawBackground  drawCheck | drawDecoration  drawDisplay  drawFocus  editorEvent  eventFilter  hasClipping  isSignalConnected | itemEditorFactory  paint  receivers  sender  senderSignalIndex  setClipping  setEditorData | setItemEditorFactory  setModelData  sizeHint  timerEvent  updateEditorGeometry |

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStyledItemDelegate

【继承关系】QtWidgets.QStyledItemDelegate –> QtWidgets.QAbstractItemDelegate –> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  connectNotify  createEditor  customEvent  disconnectNotify  displayText | editorEvent  eventFilter  initStyleOption  isSignalConnected  itemEditorFactory  paint | receivers  sender  senderSignalIndex  setEditorData  setItemEditorFactory  setModelData | sizeHint  timerEvent  updateEditorGeometry |

【详细描述】视图组件使用的默认代理类；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFileSystemModel

【继承关系】QtWidgets.QFileSystemModel -> QtWidgets.QAbstractItemModel -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| beginInsertColumns  beginInsertRows  beginMoveColumns  beginMoveRows  beginRemoveColumns  beginRemoveRows  beginResetModel  canFetchMore  changePersistentIndex  changePersistentIndexList  childEvent  columnCount  connectNotify  createIndex  customEvent  data  decodeData  directoryLoaded  disconnectNotify  dropMimeData  encodeData | endInsertColumns  endInsertRows  endMoveColumns  endMoveRows  endRemoveColumns  endRemoveRows  endResetModel  event  fetchMore  fileIcon  fileInfo  fileName  filePath  fileRenamed  filter  flags  hasChildren  headerData  iconProvider  index  isDir | isReadOnly  isSignalConnected  lastModified  mimeData  mimeTypes  mkdir  myComputer  nameFilterDisables  nameFilters  options  parent  permissions  persistentIndexList  receivers  remove  resetInternalData  resolveSymlinks  rmdir  rootDirectory  rootPath  rootPathChanged | rowCount  sender  senderSignalIndex  setData  setFilter  setIconProvider  setNameFilterDisables  setNameFilters  setOption  setOptions  setReadOnly  setResolveSymlinks  setRootPath  sibling  size  sort  supportedDropActions  testOption  timerEvent  type |

【详细描述】为本机的文件系统提供一个数据模型，可用于访问本机的文件系统，通常与 QTreeView 组合成 Model/View 结构；

setRootPath：为QFileSystemModel设置一个根目录，其参数为QDir

isDir：判断节点是不是一个目录，参数为QModelIndex对象；

filePath：返回节点的目录名或带路径的文件名，参数为QModelIndex对象；

filename：返回去除路径的文件夹或文件名，参数为QModelIndex对象；

type：返回描述节点类型的文字，如硬盘符是Drive，文件夹是FileFolder，文件则用具体的后缀描述，如text file， exe file等，参数为QModelIndex对象；

size：如果节点是文件，则返回文件大小的字节数，如果节点是文件夹，则返回0；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QEvent

【继承关系】QtCore.QEvent

【继承者】QGraphicsSceneEvent, QGraphicsSceneWheelEvent, QGraphicsSceneResizeEvent, QGraphicsSceneMoveEvent, QGraphicsSceneMouseEvent, QGraphicsSceneHoverEvent, QGraphicsSceneHelpEvent, QGraphicsSceneDragDropEvent, QGraphicsSceneContextMenuEvent, QGestureEvent, QWindowStateChangeEvent, QWhatsThisClickedEvent, QToolBarChangeEvent, QStatusTipEvent, QShowEvent, QShortcutEvent, QScrollPrepareEvent, QScrollEvent, QResizeEvent, QPaintEvent, QMoveEvent, QInputMethodQueryEvent, QInputMethodEvent, QInputEvent, QPointerEvent, QTouchEvent, QSinglePointEvent, QWheelEvent, QTabletEvent, QNativeGestureEvent, QMouseEvent, QHoverEvent, QEnterEvent, QKeyEvent, QContextMenuEvent, QIconDragEvent, QHideEvent, QHelpEvent, QFocusEvent, QFileOpenEvent, QExposeEvent, QDropEvent, QDragMoveEvent, QDragEnterEvent, QDragLeaveEvent, QCloseEvent, QActionEvent, QTimerEvent, QDynamicPropertyChangeEvent, QChildEvent

|  |  |  |  |
| --- | --- | --- | --- |
| accept  ignore | isAccepted  registerEventType | setAccepted  spontaneous | type |

【详细描述】

accpet：表示事件接收者接受此事件，对此事件进行处理，接受的事件不会再传播给上层容器组件；

ignore：表示事件接受者忽略此事件，忽略的事件将传播给长层容器组件；

type：返回事件的类型，QEventType；

QEvent.Type对象：1.ActionAdded；2.ActionChanged；3.ActionRemoved；4.ActivationChange；5.ApplicationActivate；6.ApplicationActivated；7.ApplicationDeactivate；8.ApplicationDeactivated；9.ApplicationFontChange；10.ApplicationLayoutDirectionChange；11.ApplicationPaletteChange；12.ApplicationStateChange；13.ApplicationWindowIconChange；14.ChildAdded；15.ChildPolished；16.ChildRemoved；17.Clipboard；18.Close；19.CloseSoftwareInputPanel；20.ContentsRectChange；21.ContextMenu；22.CursorChange；23.DeferredDelete；24.DragEnter；25.DragLeave；26.DragMove；27.Drop；28.DynamicPropertyChange；29.EnabledChange；30.Enter，表示鼠标光标移入事件；31.EnterWhatsThisMode；32.Expose；33.FileOpen；34.FocusAboutToChange；35.FocusIn；36.FocusOut；37.FontChange；38.Gesture；39.GestureOverride；40.GrabKeyboard；41.GrabMouse；42.GraphicsSceneContextMenu；43.GraphicsSceneDragEnter；44.GraphicsSceneDragLeave；45.GraphicsSceneDragMove；46.GraphicsSceneDrop；47.GraphicsSceneHelp；48.GraphicsSceneHoverEnter；49.GraphicsSceneHoverLeave；50.GraphicsSceneHoverMove；51.GraphicsSceneMouseDoubleClick；52.GraphicsSceneMouseMove；53.GraphicsSceneMousePress；54.GraphicsSceneMouseRelease；55.GraphicsSceneMove；56.GraphicsSceneResize；57.GraphicsSceneWheel；58.Hide；59.HideToParent；60.HoverEnter；61.HoverLeave；62.HoverMove；63.IconDrag；64.IconTextChange；65.InputMethod；66.InputMethodQuery；67.KeyboardLayoutChange；68.KeyPress；69.KeyRelease；70.LanguageChange；71.LayoutDirectionChange；72.LayoutRequest；73.Leave，表示鼠标光标移出事件；74.LeaveWhatsThisMode；75.LocaleChange；76.MacSizeChange；77.MaxUser；78.MetaCall；79.ModifiedChange；80.MouseButtonDblClick；81.MouseButtonPress，表示鼠标按键按下事件；82.MouseButtonRelease，表示鼠标按键释放事件；83.MouseMove；84.MouseTrackingChange；85.Move；86.NonClientAreaMouseButtonDblClick；87.NonClientAreaMouseButtonPress；88.NonClientAreaMouseButtonRelease；89.NonClientAreaMouseMove；90.None\_；91.OkRequest；92.OrientationChange；93.Paint；94.PaletteChange；95.ParentAboutToChange；96.ParentChange；97.PlatformPanel；98.PlatformSurface；99.Polish；100.PolishRequest；101.QueryWhatsThis；102.ReadOnlyChange；103.RequestSoftwareInputPanel；104.Resize；105.Scroll；106.ScrollPrepare；107.Shortcut；108.ShortcutOverride；109.Show；110.ShowToParent；111.SockAct；112.StateMachineSignal；113.StateMachineWrapped；114.StatusTip；115.StyleChange；116.TabletEnterProximity；117.TabletLeaveProximity；118.TabletMove；119.TabletPress；120.TabletRelease；121.TabletTrackingChange；122.ThreadChange；123.Timer；124.ToolBarChange；125.ToolTip；126.ToolTipChange；127.TouchBegin；128.TouchCancel；129.TouchEnd；130.TouchUpdate；131.UngrabKeyboard；132.UngrabMouse；133.UpdateLater；134.UpdateRequest；135.User；136.WhatsThis；137.WhatsThisClicked；138.Wheel；139.WindowActivate；140.WindowBlocked；141.WindowDeactivate；142.WindowIconChange；143.WindowStateChange；144.WindowTitleChange；145.WindowUnblocked；146.WinEventAct；147.WinIdChange；148.ZOrderChange；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMouseEvent

【继承关系】QtGui.QMouseEvent –> QtGui.QSinglePointEvent -> QtGui.QPointEvent -> Qt.Gui.QInputEvent -> QtCore.QEvent

|  |  |  |  |
| --- | --- | --- | --- |
| button  buttons  flags  globalPos | globalX  globalY  localPos  pos | screenPos  source  windowPos  x | y |

【详细描述】

button：返回Qt.MouseButton对象，表示被按下的是鼠标的那个按钮；

buttons：返回Qt.MouseButton对象的取值组合，用于判断多个按钮被按下的情况；

x/y：返回鼠标光标在接受此事件的组件上的相对坐标int；pos返回Qpoint类型；

windowPos：返回鼠标光标在接收此事件的窗口上的相对坐标，返回QPointF类型；

globalX/globalY：返回鼠标坐标在屏幕坐标上的值，返回Qpoint类型；

screenPos：返回水表光标在接收此事件的屏幕上的全局坐标，返回QPointF类型；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QKeyEvent

【继承关系】QtGui. QKeyEvent -> Qt.Gui.QInputEvent -> QtCore.QEvent

|  |  |  |  |
| --- | --- | --- | --- |
| count  isAutoRepeat  key | matches  modifiers  nativeModifiers | nativeScanCode  nativeVirtualKey  text |  |

【详细描述】

key：返回被按下的按键Qt.Key对象；

modifiers：返回Qt.kKeyboardModifier的取值组合，表示一些用于组合的按钮；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDropEvent

【继承关系】QtGui.QDropEvent -> QtCore.Event

【继承者】QDragEnterEvent, QDragMoveEvent

|  |  |  |  |
| --- | --- | --- | --- |
| acceptProposedAction  dropAction  keyboardModifiers | mimeData  mouseButtons  pos | posF  possibleActions  proposedAction | setDropAction  source |

【详细描述】放下事件。在完成拖放操作时发送该事件，即当用户在部件上放下一个对象时，发送此事件。要使部件能接收到该事件，则该部件必须接受QDragEnterEvent事件，且不能忽略QDragMoveEvnt事件。

mimeData：用于返回QDropEvent的QMimeData对象；

acceptProposedAction：表示事件接受拖动操作，允许后续的放置操作；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDragEnterEvent

【继承关系】QtGui.QDragEnterEvent -> QtGui.QDragMoveEvent -> QtGui.QDropEvent -> QtCore.Event

【详细描述】拖动进入事件。当拖动操作进入部件时，该事件被发送到部件，忽略该事件，将会导至后续的拖放事件不能被发送，此时在该部件上光标通常会在外观上显示为禁用的图形。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDragMoveEvent

【继承关系】QtGui.QDragMoveEvent -> QtGui.QDropEvent -> QtCore.Event

【详细描述】拖动移动事件。当拖动操作正在进行时，以及当具有焦点时按下键盘的修饰键(比如Ctrl)时，发送该事件，要使部件能接收到该事件，则该部件必须接受QDragEnterEvent事件。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDragLeaveEvent

【继承关系】QtGui. QDragLeaveEvent -> QtCore.Event

【详细描述】当拖放操作离开部件时发送该事件，注意：要使部件能接收到该事件，必须要使拖动先进入该部件(即产生QDragEnterEvent事件)，然后再离开该部件，才会产生QDragLeaveEvent事件。因很少使用该事件，因此本文不做重点介绍。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMimeData

【继承关系】QtCore.QMimeData -> QtCore.QObject

|  |  |  |  |
| --- | --- | --- | --- |
| retrieveData  removeFormat  clear  formats  hasFormat  setData | data  hasColor  setColorData  colorData  hasImage  setImageData | imageData  hasHtml  setHtml  html  hasText  setText | text  hasUrls  setUrls  urls |

【详细描述】QMimeData类提为数据提供一个容器，用来记录关于mime类型数据的信息，记录了一些拖动操作数据源的一些关键信息；

setText：设置文本数据；text：返回文本数据；hasText：如果对象包含文本数据，则返回True，否则返回False。

setHtml：以HTML格式设置文本数据；html：返回HTML格式的文本数据；hasHtml：如果对象包含HTML格式的文本数据，则返回True，否则为False；

setUrls：设置地址列表，参数为QUrl类的列表；urls：返回QUrl类型的地址列表；hasUrls：如果对象包Url地址，则返回True，否则为False；

setlmageData：设置图像数据，可以是QImage或QPixmap实例；imageData：返回图像对象；haslmage：如果对象包含图像数据，则返回True，否则为False.

setData：设置任意类型的MIME类型数据；参数1为QString类型，指定MIME类型；参数2为QByteData类实例。该方法可以用MIME不同类型，多次调用；data：返回mimetype类型的QByteData数据对象；hasFormat：如果对象包含mimetype类型数据，则返回True，否则为False.

formats：返回QMimeData对象中包括的所有类型列表；removeFormat：删除mimetype类型的数据；clear：删除所有数据。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QUrl

【继承关系】QtCore.QUrl

|  |  |  |  |
| --- | --- | --- | --- |
| adjusted  authority  clear  detach  errorString  fileName  fragment  fromAce  fromEncoded  fromLocalFile  fromPercentEncoding  fromStringList  fromUserInput  hasFragment | hasQuery  host  idnWhitelist  isDetached  isEmpty  isLocalFile  isParentOf  isRelative  isValid  matches  password  path  port  query | resolved  scheme  setAuthority  setFragment  setHost  setIdnWhitelist  setPassword  setPath  setPort  setQuery  setScheme  setUrl  setUserInfo  setUserName | swap  toAce  toDisplayString  toEncoded  toLocalFile  toPercentEncoding  topLevelDomain  toString  toStringList  url  userInfo  userName |

【详细描述】QUrl 类提供了一个方便的接口使用URLs。它可以解析和构造编码和未编码形式的URLs，初始化QUrl("https://github.com/")，参数为QString类型；

setUrl：构造QUrl；

path：返回对象的URL路径；

fileName：返回对象的文件名；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QRect

【继承关系】QtCore.QRect

|  |  |  |  |
| --- | --- | --- | --- |
| adjust  adjusted  bottom  bottomLeft  bottomRight  center  contains  contains  contains  contains  getCoords  getRect  height  intersected  intersects  isEmpty | isNull  isValid  left  marginsAdded  marginsRemoved  moveBottom  moveBottomLeft  moveBottomRight  moveCenter  moveLeft  moveRight  moveTo  moveTo  moveTop  moveTopLeft  moveTopRight | normalized  right  setBottom  setBottomLeft  setBottomRight  setCoords  setHeight  setLeft  setRect  setRight  setSize  setTop  setTopLeft  setTopRight  setWidth  setX | setY  size  top  topLeft  topRight  translate  translate  translated  translated  transposed  united  width  x  y |

【详细描述】QRect类定义了一个以（x，y）为左上角、给定宽度w和高度h的矩形，初始化QRect(x[int]，y[int]，w[int]，h[int])；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSize

【继承关系】

|  |  |  |  |
| --- | --- | --- | --- |
| boundedTo  expandedTo  grownBy  height | isEmpty  isNull  isValid  scale | scaled  setHeight  setWidth  shrunkBy | toTuple  transpose  transposed  width |

【详细描述】QSize类定义了使用整数精度的二维对象的大小，初始化QSize(w[int]，h[int])

setHeight：设置高度；setWidth：设置宽度；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSizePolicy

【继承关系】QtWidgets.QSizePolicy

|  |  |  |  |
| --- | --- | --- | --- |
| controlType  expandingDirections  hasHeightForWidth  hasWidthForHeight  horizontalPolicy | horizontalStretch  retainSizeWhenHidden  setControlType  setHeightForWidth  setHorizontalPolicy | setHorizontalStretch  setRetainSizeWhenHidden  setVerticalPolicy  setVerticalStretch  setWidthForHeight | transpose  transposed  verticalPolicy  verticalStretch |

【详细描述】尺寸策略，用于描述布局水平和垂直方向调整策略的属性，初始化QSizePolicy(horizontal, vertical[, type=DefaultType])

setHorizontalStretch/setVerticalStretch：设置水平/垂直大小策略的缩放因子，取值范围[0,255]；

setHeightForWidth：设置标志判断窗口部件的首选高度是否依赖于它的宽度，

QSizePolicy.PolicyFlag对象：1. QSizePolicy.GrowFlag，部件可以在必要时增长至超过它的尺寸。；2. QSizePolicy.ExpandFlag；3. QSizePolicy.ShrinkFlag；4. QSizePolicy.IgnoreFlag

QSizePolicy.Policy对象：1. QSizePolicy.Fixed，控件具有其SizeHint的尺寸且尺寸不会改变；2. QSizePolicy.Minimum，以sizeHint值是最小值，尺寸可以变大，但不能比sizeHint保存的值更小；3. QSizePolicy.Maximum，以sizeHint为最大值，尺寸可以被挤压到任意小，但不能比sizeHint保存的值更大；4. QSizePolicy.Preferred，以sizeHint为最佳选择，但是根据情况也可以变大缩小；5. QSizePolicy.Expanding，以sizeHint为合理的大小，但是会尽可能为控件提供更大的空间；6. QSizePolicy.MinimumExpanding，以sizeHint为最小尺寸，但是会尽可能获得更大的空间；7. QSizePolicy.Ignored，忽略sizeHin值，尽可能获得更大的空间；

QSizePolicy.ControlType对象：1. QSizePolicy.DefaultType；2. QSizePolicy.ButtonBox；3. QSizePolicy.CheckBox；4. QSizePolicy.ComboBox；5. QSizePolicy.Frame；6. QSizePolicy.GroupBox；7. QSizePolicy.Label；8. QSizePolicy.Line；9. QSizePolicy.LineEdit；10. QSizePolicy.PushButton；11. QSizePolicy.RadioButton；12. QSizePolicy.Slider；13. QSizePolicy.SpinBox；14. QSizePolicy.TabWidget；15. QSizePolicy.ToolButton；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QIcon

【继承关系】QtGui.QIcon

|  |  |  |  |
| --- | --- | --- | --- |
| actualSize  addFile  addPixmap  availableSizes  cacheKey  fallbackSearchPaths | fallbackThemeName  fromTheme  hasThemeIcon  isDetached  isMask  isNull | name  paint  pixmap  setFallbackSearchPaths  setFallbackThemeName  setIsMask | setThemeName  setThemeSearchPaths  swap  themeName  themeSearchPaths |

【详细描述】图标对象，初始化1. QIcon()构造一个空图像构成的图标；2. QIcon(pixmap[QPixmap])从Pixmap对象构造函数；3. QIcon(filename[QString])从图像文件构造图标；

addFile：从图像文件中向 QIcon 对象添加图像，参数1. filename[QString]；2. size[QSize]；3. mode，枚举值，QIcon.Mode对象，设置使用的模式；4. State，枚举值，QIcon.State对象；

addPixmap：将 QPixmap对象添加到QIcon对象中，参数1. pixmap[QPixmap]；2. mode，枚举值，QIcon.Mode对象，设置使用的模式；3. State，枚举值，QIcon.State对象；

QIcon.Mode对象：1. QIcon.Normal，窗口为使能状态，但未激活；2. QIcon.Disabled，窗口为禁用状态；3. QIcon.Active，窗口为激活状态；4. QIcon.Selected，窗口被选中；

QIcon.State对象：1. QIcon.Off，窗口在关状态；2. QIcon.On，窗口在开状态；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QToolTip

|  |  |  |  |
| --- | --- | --- | --- |
| font  hideText | isVisible  palette | setFont  setPalette | showText  text |

【详细描述】

setFont：设置提示气泡提示信息的字体与字号大小；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPalette

|  |  |  |  |
| --- | --- | --- | --- |
| alternateBase  base  brightText  brush  button  buttonText  cacheKey  color  currentColorGroup | dark  highlight  highlightedText  isBrushSet  isCopyOf  isEqual  light  link  linkVisited | mid  midlight  placeholderText  resolve  setBrush  setColor  setColorGroup  setCurrentColorGroup  shadow | swap  text  toolTipBase  toolTipText  window  windowText |

【详细描述】调色板，它管理着控件或窗体的所有颜色信息，每个窗体或控件都包含一个QPalette对象，在显示时按照它的QPalette对象中对各部分各状态下的颜色的描述来进行绘制。

setColor：设置xx的颜色，第一个参数为QPalette对象，第二个参数为颜色；

window：返回窗口部件的背景对象；

WindowText：返回窗口部件的前景色对象

base：返回文本部件背景色对象

text：返回文本部件的前景色对象

highlight：返回文本部件选中的背景色对象

highlightedText：返回文本部件选中的文本颜色对象

button：返回按钮部件背景色【注意：QPushButton的背景色涉及样式表，所以不能通过QPalette修改】

ToolTipBase：返回QToolTip背景色对象

toolTipText：返回QToolTip前景色对象

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDir

【继承关系】QtCore.QDir

|  |  |  |  |
| --- | --- | --- | --- |
| absoluteFilePath  absolutePath  addSearchPath  canonicalPath  cd  cdUp  cleanPath  count  current  currentPath  dirName  drives  entryInfoList  entryList  exists | filePath  filter  fromNativeSeparators  home  homePath  isAbsolute  isAbsolutePath  isEmpty  isReadable  isRelative  isRelativePath  isRoot  listSeparator  makeAbsolute  match | mkdir  mkpath  nameFilters  nameFiltersFromString  path  refresh  relativeFilePath  remove  removeRecursively  rename  rmdir  rmpath  root  rootPath  searchPaths | separator  setCurrent  setFilter  setNameFilters  setPath  setSearchPaths  setSorting  sorting  swap  temp  tempPath  toNativeSeparators |

【详细描述】

currentPath：用于获取应用程序的当前路径

filter对象：Dirs，列出匹配filters的目录；AllDirs，列出所有目录；Files，列出所有文件；Drives，列出磁盘驱动器（Unix系统忽略）；NoSymLinks，不列出符号连接；NoDotAndDotDot，不列出特殊条目，如“.”或者“..”；NoDot，不列出“.”；NoDotDot，不列出“..”；AllEntries，列出目录、驱动盘、符号连接；Readable，列出程序有读权限的文件。可读值需配合Dirs和Files；Writable，列出程序有写权限的文件。可读值需配合Dirs和Files；Executable，列出程序有运行权限的文件。可读值需配合Dirs和Files；Modified，只列出已修改的文件（Unix系统忽略）；Hidden，列出系统文件；CaseSensitive，过滤器应区分大小写

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QPen

setWidth：设置线条宽度；

setColor：设置画线的颜色，参数为参数为枚举值，Qt.GlobalColor对象；

setStyle：设置线的类型，如实线，虚线等，参数为参数为枚举值，Qt.PenStyle对象；

setCapStyle：设置线端点样式，参数为参数为枚举值，Qt.PenCapStyle对象；

setJoinStyle：设置线得连接点样式，参数为参数为枚举值，Qt.PenJoinStyle对象；

setDashOffset：

setDashPattern：

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QBrush

setColor: 设置画刷的颜色，参数为参数为枚举值，Qt.GlobalColor对象；

setStyle：设置填充样式，参数为参数为枚举值，Qt.BrushStyle对象；

setTexture：设置一个QPixmap类型的图片pixmap作为画刷的图片，画刷样式自动在设置为Qt.TexturePattern对象；

setTextureImage：设置一个QImage类型的图片image作为画刷的图片，画刷样式自动在设置为Qt.TexturePattern对象；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QDate, QTime, QDateTime

【继承关系】QtCore.QDate

|  |  |  |  |
| --- | --- | --- | --- |
| addDays  addMonths  addYears  currentDate  day  dayOfWeek  dayOfYear  daysInMonth | daysInYear  daysTo  endOfDay  fromJulianDay  fromString  getDate  isLeapYear  isNull | isValid  longDayName  longMonthName  month  setDate  shortDayName  shortMonthName  startOfDay | toJulianDay  toPyDate  toString  weekNumber  year |

【继承关系】QtCore.QTime

|  |  |  |  |
| --- | --- | --- | --- |
| addMSecs  addSecs  currentTime  elapsed  fromMSecsSinceStartOfDay | fromString  hour  isNull  isValid  minute | msec  msecsSinceStartOfDay  msecsTo  restart  second | secsTo  setHMS  start  toPyTime  toString |

【继承关系】QtCore.QDateTime

|  |  |  |  |
| --- | --- | --- | --- |
| addDays  addMonths  addMSecs  addSecs  addYears  currentDateTime  currentDateTimeUtc  currentMSecsSinceEpoch  currentSecsSinceEpoch  date  daysTo | fromMSecsSinceEpoch  fromSecsSinceEpoch  fromString  fromTime\_t  isDaylightTime  isNull  isValid  msecsTo  offsetFromUtc  secsTo  setDate | setMSecsSinceEpoch  setOffsetFromUtc  setSecsSinceEpoch  setTime  setTime\_t  setTimeSpec  setTimeZone  swap  time  timeSpec  timeZone | timeZoneAbbreviation  toLocalTime  toMSecsSinceEpoch  toOffsetFromUtc  toPyDateTime  toSecsSinceEpoch  toString  toTime\_t  toTimeSpec  toTimeZone  toUTC |

【详细描述】PyQt5用于处理日期和时间的类有QDate，QDateTime，QTime。QDate类用于处理公历日期。它提供的方法可以测定日期、比较或者操作日期。QTime类用于处理时间。它提供的方法可以测定时间、比较或者多样化操作时间。QDateTime类将QDate对象和QTime对象整合为一个对象。

currentDate：获取当前系统时间；

daysTo：返回时间间隔；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QVariant

【详细描述】这个类型相当于是Java里面的Object，它把绝大多数Qt提供的数据类型都封装起来，起到一个数据类型“擦除”的作用。比如我们的 table单元格可以是string，也可以是int，也可以是一个颜色值，那么这么多类型怎么返回呢？于是，Qt提供了这个QVariant类型，你可以把这很多类型都存放进去，到需要使用的时候使用一系列的to函数取出来即可。比如你把int包装成一个QVariant，使用的时候要用QVariant.toInt()重新取出来。这里需要注意的是，QVariant类型的放入和取出必须是相对应的，你放入一个int就必须按int取出，不能用toString()，Qt不会帮你自动转换。

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFont

【继承关系】QGui.QFont

|  |  |  |  |
| --- | --- | --- | --- |
| bold  cacheStatistics  capitalization  cleanup  defaultFamily  exactMatch  families  family  fixedPitch  fromString  hintingPreference  initialize  insertSubstitution  insertSubstitutions  isCopyOf  italic  kerning | key  lastResortFamily  lastResortFont  letterSpacing  letterSpacingType  overline  pixelSize  pointSize  pointSizeF  rawMode  rawName  removeSubstitutions  resolve  setBold  setCapitalization  setFamilies  setFamily | setFixedPitch  setHintingPreference  setItalic  setKerning  setLetterSpacing  setOverline  setPixelSize  setPointSize  setPointSizeF  setRawMode  setRawName  setStretch  setStrikeOut  setStyle  setStyleHint  setStyleName  setStyleStrategy | setUnderline  setWeight  setWordSpacing  stretch  strikeOut  style  styleHint  styleName  styleStrategy  substitute  substitutes  substitutions  swap  toString  underline  weight  wordSpacing |

【详细描述】

setPixelSize：设置字体高度占用的像素px大小。

setPointSize：设置字体高度占用的可物理测量的长度磅（point）大小，一般用于印刷领域。

setFamily：设置字体；

setBold：设置字体加粗；

setItalic：设置斜体；

setOverline：设置上划线；

setUnderline：设置下划线；

setStrikeOut：设置删除线；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QLayout

【继承关系】QtWidgets.QLayout -> QtCore.QObject/ QtWidgets.QLayoutItem

|  |  |  |  |
| --- | --- | --- | --- |
| activate  addChildLayout  addChildWidget  addItem  addWidget  alignmentRect  childEvent  closestAcceptableSize  connectNotify  contentsMargins  contentsRect  controlTypes  count | customEvent  disconnectNotify  expandingDirections  geometry  getContentsMargins  indexOf  invalidate  isEmpty  isEnabled  isSignalConnected  itemAt  layout  maximumSize | menuBar  minimumSize  parentWidget  receivers  removeItem  removeWidget  replaceWidget  sender  senderSignalIndex  setAlignment  setContentsMargins  setEnabled  setGeometry | setMenuBar  setSizeConstraint  setSpacing  sizeConstraint  spacing  takeAt  timerEvent  totalHeightForWidth  totalMaximumSize  totalMinimumSize  totalSizeHint  update  widgetEvent |

【详细描述】

setContentsMargins：设置四个边的边距，参数为layoutLeftMargin【int】，layoutTopMargin【int】，layoutRightMargin【int】，layoutBottomMargin【int】分别表示左、上、右、下的外边距；

setSizeConstraint：设置尺寸约束，其参数为枚举值，QLayout.SizeConstraint对象

setMargin：设置左、上、右、下的外边距，但是他们的外边距是相同的，参数为int；

setSpacing：设置各个控件之间的间距，参数为int；

QLayout.SizeConstraint对象：1. SetDefaultConstraint；2. SetFixedSize；3. SetMinimumSize；4. SetMaximumSize；5. SetMinAndMaxSize；6. SetNoConstraint

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QBoxLayout

【继承关系】QtWidgets.QBoxLayout -> QtWidgets.QLayout -> QtCore.QObject/ QtWidgets.QLayoutItem

|  |  |  |  |
| --- | --- | --- | --- |
| addChildLayout  addChildWidget  addItem  addLayout  addSpacerItem  addSpacing  addStretch  addStrut  addWidget  alignmentRect  childEvent  connectNotify | count  customEvent  direction  disconnectNotify  expandingDirections  hasHeightForWidth  heightForWidth  insertItem  insertLayout  insertSpacerItem  insertSpacing  insertStretch | insertWidget  invalidate  isSignalConnected  itemAt  maximumSize  minimumHeightForWidth  minimumSize  receivers  sender  senderSignalIndex  setDirection  setGeometry | setSpacing  setStretch  setStretchFactor  sizeHint  spacing  stretch  takeAt  timerEvent  widgetEvent |

【详细描述】

setStretch：将布局内索引号为index的组件延伸系数设置为stretch，参数为index, stretch；

addStretch：添加了一个伸缩空间（QSpacerItem）

setStretchFactor：设置控件、布局的拉伸系数；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QHBoxLayout/QVBoxLayout

【继承关系】QtWidgets.QHBoxLayout/ QVBoxLayout -> QtWidgets.QBoxLayout -> QtWidgets.QLayout -> QtCore.QObject/ QtWidgets.QLayoutItem

|  |  |  |  |
| --- | --- | --- | --- |
| addChildLayout  addChildWidget  alignmentRect  childEvent | connectNotify  customEvent  disconnectNotify  isSignalConnected | receivers  sender  senderSignalIndex  timerEvent | widgetEvent |

【详细描述】垂直布局，水平布局

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QGridLayout

【继承关系】QtWidgets. QGridLayout -> QtWidgets.QLayout -> QtCore.QObject/ QtWidgets.QLayoutItem

|  |  |  |  |
| --- | --- | --- | --- |
| addChildLayout  addChildWidget  addItem  addLayout  addWidget  alignmentRect  cellRect  childEvent  columnCount  columnMinimumWidth  columnStretch  connectNotify  count | customEvent  disconnectNotify  expandingDirections  getItemPosition  hasHeightForWidth  heightForWidth  horizontalSpacing  invalidate  isSignalConnected  itemAt  itemAtPosition  maximumSize  minimumHeightForWidth | minimumSize  originCorner  receivers  rowCount  rowMinimumHeight  rowStretch  sender  senderSignalIndex  setColumnMinimumWidth  setColumnStretch  setDefaultPositioning  setGeometry  setHorizontalSpacing | setOriginCorner  setRowMinimumHeight  setRowStretch  setSpacing  setVerticalSpacing  sizeHint  spacing  takeAt  timerEvent  verticalSpacing  widgetEvent |

【详细描述】栅格布局

setHorizontalSpacing/setVerticalSpacing：设置设置两个控件之间的水平和竖直距离，参数为layoutHorizontalSpacing【int】/ layoutVerticalSpacing【int】；

setRowStretch/ setColumnStretch：由于网格布局管理器中的组件会随着窗口拉伸而发生变化的，所以需要设置组件之间的比例系数，参数1. row，表示设置第0列、第1列两者在水平方向的空间比例分配

addWidget：添加组件，参数为1. widget【QWidget】；2. Row【int】/Column【int】分别表示组件所在的行号和列号；4. rowSpan【int】/5. columnSpan【int】分别表示组件所占的行数和列数；

addLayout：用于在布局中插入子布局，参数为1. layout， 2. Row/ Column分别表示组件所在的行号和列号

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QGroupBox

【继承关系】QtWidgets.QGroupBox -> QtWidgets. QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  alignment  changeEvent  childEvent  clicked  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent | dropEvent  enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  hideEvent  initPainter  initStyleOption  inputMethodEvent  isCheckable  isChecked  isFlat | isSignalConnected  keyPressEvent  keyReleaseEvent  leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent | sender  senderSignalIndex  setAlignment  setCheckable  setChecked  setFlat  setTitle  sharedPainter  showEvent  tabletEvent  timerEvent  title  toggled  updateMicroFocus  wheelEvent |

【详细描述】

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFrame

【继承关系】QtWidgets.QFrame –> QtWidgets. QWidget

【继承者】QChartView，QHelpContentWidget，QHelpIndexWidget，QAbstractItemView，QAbstractScrollArea，QColumnView，QGraphicsView，QHeaderView，QLCDNumber，QLabel，QListView，QListWidget，QMdiArea，QPlainTextEdit，QScrollArea，QSplitter，QStackedWidget，QTableView，QTableWidget，QTextBrowser，QTextEdit，QToolBox，QTreeView，QTreeWidget，QUndoView；

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  changeEvent  childEvent  closeEvent  connectNotify  contextMenuEvent  create  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent  enterEvent | event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  frameRect  frameShadow  frameShape  frameStyle  frameWidth  hideEvent  initPainter  initStyleOption  inputMethodEvent  isSignalConnected | keyPressEvent  keyReleaseEvent  leaveEvent  lineWidth  metric  midLineWidth  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  nativeEvent  paintEvent  receivers  resizeEvent  sender | senderSignalIndex  setFrameRect  setFrameShadow  setFrameShape  setFrameStyle  setLineWidth  setMidLineWidth  sharedPainter  showEvent  sizeHint  tabletEvent  timerEvent  updateMicroFocus  wheelEvent |

【详细描述】QFrame类继承自QWidget类，在组件周围添加了边框。 这个类又派生出一些组件，如标签，多行文本字段等。

setFrameShape：设置QFrame类窗口的边框，参数为枚举值，QFrame.Shape对象；

setFrameShadow：设置QF rame类窗口的阴影，参数为枚举值，QFrame.Shadow对象；

setFrameStyle：同时设置边框和阴影，参数为枚举值，QFrame.Shape和QFrame.Shadow的"|"组合；

QFrame.Shape对象：1. QFrame.NoFrame，无边框；2. QFrame.Box，矩形框；3. QFrame.Panel，凸起或凹陷的面板；4. QFrame.WinPanel，Windows 2000风格的面板；5. QFrame.HLine，水平线(用作分隔符)；6. QFrame.VLine，垂直线(用作分隔符)；7. QFrame.StyledPanel，依据当前GUI类型，画一个矩形面板，可以凸起或下沉；

QFrame.Shadow对象：1. Plain，无阴影；2. Raised，面板凸起；3. Sunken，面板下沉；#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

QSpacerItem

【继承关系】QtWidgets.QSpacerItem –> QtWidgets.QLayoutItem

|  |  |  |  |
| --- | --- | --- | --- |
| changeSize  expandingDirections  geometry | isEmpty  maximumSize  minimumSize | setGeometry  sizeHint  sizePolicy | spacerItem |

【详细描述】QSpacerItem为布局提供了一个空白区，初始化QSpacerItem (w, h[, hData=QSizePolicy.Minimum[, vData=QSizePolicy.Minimum]] )

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QSplitter

【继承关系】QtWidgets. QSplitter -> QtWidgets.QFrame –> QtWidgets. QWidget

|  |  |  |  |
| --- | --- | --- | --- |
| actionEvent  addWidget  changeEvent  childEvent  childrenCollapsible  closeEvent  closestLegalPosition  connectNotify  contextMenuEvent  count  create  createHandle  customEvent  destroy  disconnectNotify  dragEnterEvent  dragLeaveEvent  dragMoveEvent  drawFrame  dropEvent | enterEvent  event  focusInEvent  focusNextChild  focusNextPrevChild  focusOutEvent  focusPreviousChild  getRange  handle  handleWidth  hideEvent  indexOf  initPainter  initStyleOption  inputMethodEvent  insertWidget  isCollapsible  isSignalConnected  keyPressEvent  keyReleaseEvent | leaveEvent  metric  minimumSizeHint  mouseDoubleClickEvent  mouseMoveEvent  mousePressEvent  mouseReleaseEvent  moveEvent  moveSplitter  nativeEvent  opaqueResize  orientation  paintEvent  receivers  refresh  replaceWidget  resizeEvent  restoreState  saveState  sender | senderSignalIndex  setChildrenCollapsible  setCollapsible  setHandleWidth  setOpaqueResize  setOrientation  setRubberBand  setSizes  setStretchFactor  sharedPainter  showEvent  sizeHint  sizes  splitterMoved  tabletEvent  timerEvent  updateMicroFocus  wheelEvent  widget |

【详细描述】分割窗口QSplitter类，用于实现分割条功能，可以左右分割或上下分割，一般是在两个可以自由改变大小的组件之间进行分割；

setOrientation：设置QSplitter的方向，参数为枚举值，Qt.Orientation对象

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QStackedWidget

【继承关系】QtWidgets.QStackedWidget -> QtWidgets.Qframe -> QtWidgets.QWidget -> QtCore.QObject/ QtGui.QPaintDevice

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## Qt对象

【继承关系】QtCore.Qt

|  |  |  |  |
| --- | --- | --- | --- |
| AlignmentFlag  AnchorPoint  ApplicationAttribute  ApplicationState  ArrowType  AspectRatioMode  Axis  BGMode  BrushStyle  CaseSensitivity  CheckState  ChecksumType  ClipOperation  ConnectionType  ContextMenuPolicy  CoordinateSystem  Corner  CursorMoveStyle  CursorShape  DateFormat  DayOfWeek | DockWidgetArea  DropAction  Edge  EnterKeyType  EventPriority  FillRule  FindChildOption  FocusPolicy  FocusReason  GestureFlag  GestureState  GestureType  GlobalColor  HighDpiScaleFactorRoundingPolicy  HitTestAccuracy  ImageConversionFlag  InputMethodHint  InputMethodQuery  ItemDataRole  ItemFlag  ItemSelectionModel | ItemSelectionOperation  Key  KeyboardModifier  LayoutDirection  MaskMode  MatchFlag  Modifier  MouseButton  MouseEventFlag  MouseEventSource  NativeGestureType  NavigationMode  Orientation  PenCapStyle  PenJoinStyle  PenStyle  ScreenOrientation  ScrollBarPolicy  ScrollPhase  ShortcutContext  SizeHint | SizeMode  SortOrder  TabFocusBehavior  TextElideMode  TextFlag  TextFormat  TextInteractionFlag  TileRule  TimerType  TimeSpec  ToolBarArea  ToolButtonStyle  TouchPointState  TransformationMode  UIEffect  WhiteSpaceMode  WidgetAttribute  WindowFrameSection  WindowModality  WindowState  WindowType |

【详细描述】

Qt.GlobalColor对象：1. Qt.white，2. Qt.black，3. Qt.red，4. Qt.darkRed，5. Qt.green，6. Qt.darkGreen，7. Qt.blue，8. Qt.darkBlue，9. Qt.cyan，10. Qt.darkCyan，11. Qt.magenta，12. Qt.darkMagenta，13. Qt.yellow，14. Qt.darkYellow，15. Qt.gray，16. Qt.darkGray，17. Qt.lightGray，18. Qt.transparent，19. Qt.color0，20. Qt.color1；

Qt.WindowType对象：1. Qt.Widget（小工具），QWidget构造函数的默认值，如新的窗口部件没有父窗口部件，则它是一个独立的窗口，否则就是一个子窗口部件；2. Qt.Window（窗口），无论是否有父窗口部件，新窗口部件都是一个窗口，通常有一个窗口边框和一个标题栏；3. Qt.Dialog（对话框）：新窗口部件是一个对话框；4. Qt.Sheet（表），新窗口部件是一个Mac表单；Qt.Drawer（抽屉），新窗口部件是一个Mac抽屉；5. Qt.Popup（弹出菜单），新窗口部件是一个弹出式顶层窗口；6. Qt.Tool（工具栏）: 新窗口部件是一个工具窗口，它通常是一个用于显示工具按钮的小窗口，如果一个工具窗口有父窗口部件，则它将显示在父窗口部件的上面，否则，将相当于使用了Qt.WindowStaysOnTopHint展示；7. Qt.Tooltip（提示），新窗口部件是一个提示窗口，没有标题栏和窗口边框；8. Qt.SplashScreen（启动窗口），新窗口部件是一个欢迎窗口，它是QSplashScreen构造函数的默认值；9. Qt.Desktop（桌面），新窗口部件是桌面，它是QDesktopWidget构造函数的默认值；10. Qt.SubWindow（子窗口），新窗口部件是一个子窗口，而无论该窗口部件是否有父窗口部件；11. Qt.ForeignWindow(外部窗口)，用于表示窗口为另一进程创建；12. Qt.X11BypassWindowManagerHint；完全忽视窗口管理器，它的作用是产生一个根本不被管理器的无窗口边框的窗口，此时，用户无法使用键盘进行输入，除非手动调用QWidget.ActivateWindow函数。13. Qt.FramelessWindowHint，产生一个无窗口边框的窗口，此时用户无法移动该窗口和改变它的大小】；此外，通过如下设置可以为窗口添加一些功能，【14. Qt. MSWindowsFixedSizeDialogHint，禁止调整窗口尺寸；15. Qt.CustomizeWindowHint，关闭默认的窗口标题提示；16. Qt.WindowTitleHint，为窗口修饰一个标题栏；17. Qt. WindowSystemMenuHint为窗口修饰一个窗口菜单系统；18. Qt.WindowMinimizeButtonHint，为窗口添加最小化按钮；19. Qt.WindowMaximizeButtonHint，为窗口添加最大化按钮；20. Qt.WindowMinMaxButtonsHint，为窗口添加最大化和最小化按钮；21. Qt.WindowCloseButtonHint，窗口只有一个关闭按钮；22. Qt.WindowStaysOnTopHint，总在最上面的窗口,置前】；windowFlags，获取当前窗口的属性；

Qt.WindowModality对象：1. Qt.NonModal，非模态，可以和程序的其他窗口交互；2. Qt.WindowModal，窗口模态，程序在未处理外当前对话框时，将阻止和对话框的父窗口进行交互；3. Qt.ApplicaitonModal，应用程序模态，阻止和任何其他窗口进行交互；

Qt.AlignmentFlag对象：1. Qt.AlignLeft，水平方向靠左对齐；2. Qt.AlignRight，水平方向靠右对齐；3. Qt.AlignHCenter，水平方向居中对齐；4. Qt.AlignJustify，水平方向调整间距两端对齐；5. Qt.AlignTop，垂直方向靠上对齐；6. Qt.AlignBottom，垂直方向靠下对齐；7. Qt.AlignVCenter，垂直方向居中对齐；8. Qt.AlignBaseline；9. Qt.AlignCenter；10. Qt.AlignAbsolute；11. Qt.AlignLeading；12. Qt.AlignTrailing；13. Qt.AlignHorizontal\_Mask；14. Qt.AlignVertical\_Mask；

Qt.TextInteractionFlag对象：1. Qt.NoTextInteraction，不允许与文本进行交互；2. Qt.TextSelectableByMouse 可通过鼠标选择；3. TextSelectableByKeyboard，可通过鼠标键盘选择；4. Qt.LinksAccessibleByMouse：可以用鼠标高亮显示和激活链接；5. Qt.LinksAccessibleByKeyboard，可以用键盘上的回车键(enter) 高亮显示和激活链接；6. Qt.TextEditable，允许编辑；7. Qt.TextEditorInteraction，禁止使用鼠标；8. Qt.TextBrowserInteraction，允许文本交互；

Qt.CheckState对象：1. Qt.Unchecked，返回0，组件被选中；2. Qt.PartiallyChecked，返回1，组件被半选中；3. Qt.Checked，返回2，组件没有选中；

Qt.Orientation对象：1. Qt.Vertical，2. Qt.Horizontal；

Qt.PenStyle对象：1. Qt.NoPen；2. SolidLine；3. DashLine；4. DotLine；5. DashDotLine；6. DashDotDotLine；7. CustomDashLine；

Qt.PenCapStyle对象：1. Qt.FlatCap；2. Qt.SquareCap；3. Qt.RoundCap；

Qt.PenJoinStyle对象：1. Qt.MiterJoin；2. Qt.BevelJoin；3. Qt.RoundJoin；4. Qt.SvgMiterJoin；

Qt.BrushStyle对象：1. Qt.NoBrush；2. Qt.SolidPattern；3. Qt.Dense1Pattern；4. Qt.Dense2Pattern；5. Qt.Dense3Pattern；6. Qt.Dense4Pattern；7. Qt.Dense5Pattern；8. Qt.Dense6Pattern；9.Qt.Dense7Pattern；10. Qt.HorPattern；11. Qt.VerPattern；12. Qt.CrossPattern；13. Qt.BDiagPattern；14. Qt.FDiagPattern；15. Qt.DiagCrossPattern；16. Qt.LinearGradientPattern；17. Qt.ConicalGradientPattern；18. Qt.RadialGradientPatter；19. Qt.TexturePattern

Qt.DayOfWeek对象：1. Qt.Monday；2. Qt.Tuesday；3. Qt.Wednesday；4. Qt.Thursday；5. Qt.Friday；6.Qt.Saturday；7. Qt.Sunday；

Qt.MatchFlag对象：1. MatchExactly；2. MatchFixedString；3. MatchContains；4. MatchStartsWith；5. MatchEndsWith；6. MatchCaseSensitive；7. MatchRegExp；8. MatchRegularExpression；

9. MatchWildcard；10. MatchWrap；11. MatchRecursive；

Qt.SortOrder对象：1. Qt.AscendingOrder；2. Qt.DescendingOrder；

Qt.ContextMenuPolicy对象：1. Qt.NoContextMenu；2. Qt.PreventContextMenu；3. Qt.DefaultContextMenu；4. Qt.ActionsContextMenu；5. Qt.CustomContextMenu；

Qt.ItemDataRole对象：1. Qt.DisplayRole，定义角色为视图中显示的字符串；2. Qt.DecorationRole，；3. Qt.EditRole；4. Qt.ToolTipRole，控制鼠标提示信息；5. Qt.StatusTipRole；6. Qt.WhatsThisRole；7. Qt.SizeHintRole；8. Qt.FontRole，控制文字字体；9. Qt.TextAlignmentRole；10. Qt.BackgroundRole，控制显示背景；11. Qt.BackgroundColorRole；12. Qt.ForegroundRole，控制文字颜色；13. Qt.TextColorRole；14. Qt.CheckStateRole，控制复选框的状态；15. Qt.InitialSortOrderRole；16. Qt.AccessibleTextRole；17. Qt.AccessibleDescriptionRole；18. Qt.UserRole，定义角色为自定义数据；

Qt.ItemFlags对象：1. Qt.NoItemFlags，数据项没有设置属性；2. Qt.ItemIsSelectable，数据项可以被选择；3. Qt.ItemIsEditable，数据项可以被编辑；4. Qt.ItemIsDragEnabled，数据项可以被拖拽；5. Qt.ItemIsDropEnabled，数据项可以接受拖放中的释放操作，即拖拽数据可以在此处释放；6. Qt.ItemIsUserCheckable，数据项可以进行选中或未选中操作，即是否带有复选框标记；7. Qt.ItemIsEnabled，数据项支持用户交互；8. Qt.ItemIsAutoTristate，数据项的状态依赖于其子项的状态，用于QTreeWiget对象中，当所有子项都被Checked则父项自动变为checked，但所有子项都为unchecked，则父项自动变为unchecked，否则为部分选中状态；9. Qt.ItemNeverHasChildren，数据项不可能有子项；10. Qt.ItemIsUserTristate，用户可以在三种不同的状态中循环；

Qt.MouseButton对象：1. Qt.NoButton；2. Qt.AllButtons；3. Qt.LeftButton；4. Qt.RightButton；5. Qt.MiddleButton

Qt.DropAction对象：1. Qt.CopyAction，将数据复制到drop site组件；2. Qt.MoveAction，将数据从drag site组件移动到drop site组件；3. Qt.LinkAction，在drag site组件和drop site组件之间建立数据连接；4. Qt.ActionMask；5. Qt.IgnoreAction；6. Qt.TargetMoveAction；

Qt.WidgetAttribute对象：1. Qt.WA\_AcceptDrops；2. Qt.WA\_AlwaysShowToolTips；3. Qt.WA\_CustomWhatsThis；4. Qt.WA\_DeleteOnClose，用于设置窗体在关闭时是否自动删除；

Qt.WindowState对象：1. Qt.WindowNoState，正常状态；2. Qt.WindowMinimized，窗口最小化；3. Qt.WindowMaximized，窗口最大化；4. Qt.WindowFullScreen，窗口填充整个屏幕，且没有边框；5. Qt.WindowActive，变为活动的窗口，例如可以接受键盘输入；

Qt.ToolButtonStyle对象：1. Qt.ToolButtonIconOnly，只显示图标；2. Qt.ToolButtonTextBesideIcon，只显示文字；3. Qt.ToolButtonTextBesideIcon，文字在图标旁；4. Qt.ToolButtonTextUnderIcon，文字在图标下；5. Qt.ToolButtonFollowStyle，根据QStyle::StyleHint格式显示；

Qt.DockWidgetArea对象：1. Qt.LeftDockWidgetArea；2. Qt.RightDockWidgetArea；3. Qt.TopDockWidgetArea；4. Qt.BottomDockWidgetArea；5. Qt.AllDockWidgetAreas；6. Qt.NoDockWidgetArea

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QMetaType

【继承关系】QtCore.QMetaType

|  |  |  |  |
| --- | --- | --- | --- |
| flags  id  isRegistered | isValid  metaObjectForType  name | type  typeFlags  typeName |  |

【详细描述】即QVariant

QMetaType.Type对象：1.QMetaType.Bool；2.QMetaType.Char；3.QMetaType.Char16；4.QMetaType.Char32；5.QMetaType.Double；6.QMetaType.Float；7.QMetaType.Int；8.QMetaType.Long；9.QMetaType.LongLong；10.QMetaType.Nullptr；11.QMetaType.QBitArray；12.QMetaType.QBitmap；13.QMetaType.QBrush；14.QMetaType.QByteArray；15.QMetaType.QByteArrayList；16.QMetaType.QCborArray；17.QMetaType.QCborMap；18.QMetaType.QCborSimpleType；19.QMetaType.QCborValue；20.QMetaType.QChar；21.QMetaType.QColor；22.QMetaType.QColorSpace；23.QMetaType.QCursor；24.QMetaType.QDate；25.QMetaType.QDateTime；26.QMetaType.QEasingCurve；27.QMetaType.QFont；28.QMetaType.QIcon；29.QMetaType.QImage；30.QMetaType.QJsonArray；31.QMetaType.QJsonDocument；32.QMetaType.QJsonObject；33.QMetaType.QJsonValue；34.QMetaType.QKeySequence；35.QMetaType.QLine；36.QMetaType.QLineF；37.QMetaType.QLocale；38.QMetaType.QMatrix4x4；39.QMetaType.QModelIndex；40.QMetaType.QObjectStar；41.QMetaType.QPalette；42.QMetaType.QPen；43.QMetaType.QPersistentModelIndex；44.QMetaType.QPixmap；45.QMetaType.QPoint；46.QMetaType.QPointF；47.QMetaType.QPolygon；48.QMetaType.QPolygonF；49.QMetaType.QQuaternion；50.QMetaType.QRect；51.QMetaType.QRectF；52.QMetaType.QRegion；53.QMetaType.QRegularExpression；54.QMetaType.QSize；55.QMetaType.QSizeF；56.QMetaType.QSizePolicy；57.QMetaType.QString；58.QMetaType.QStringList；59.QMetaType.QTextFormat；60.QMetaType.QTextLength；61.QMetaType.QTime；62.QMetaType.QTransform；63.QMetaType.QUrl；64.QMetaType.QUuid；65.QMetaType.QVariant；66.QMetaType.QVariantHash；67.QMetaType.QVariantList；68.QMetaType.QVariantMap；69.QMetaType.QVector2D；70.QMetaType.QVector3D；71.QMetaType.QVector4D；72.QMetaType.SChar；73.QMetaType.Short；74.QMetaType.UChar；75.QMetaType.UInt；76.QMetaType.ULong；77.QMetaType.ULongLong；78.QMetaType.UnknownType；79.QMetaType.User；80.QMetaType.UShort；81.QMetaType.Void；82.QMetaType.VoidStar

QMetaType.TypeFlag对象：1. QMetaType.NeedsConstruction；2.QMetaType.RelocatableType；3. QMetaType.IsEnumeration；4. QMetaType.IsUnsignedEnumeration；5. QMetaType.PointerToQObject；6. QMetaType.IsPointer；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QIODevice

【继承关系】QtCore.QIODevice –> QtCore.QObject/QtCore.QIODeviceBase

【继承者】QNetworkReply, QLocalSocket, QAbstractSocket, QUdpSocket, QTcpSocket, QProcess, QFileDevice, QSaveFile, QFile, QTemporaryFile, QSslSocket, QBuffer

|  |  |  |  |
| --- | --- | --- | --- |
| aboutToClose  atEnd  bytesAvailable  bytesToWrite  bytesWritten  canReadLine  channelBytesWritten  channelReadyRead  childEvent  close  commitTransaction  connectNotify  currentReadChannel  currentWriteChannel  customEvent | disconnectNotify  errorString  getChar  isOpen  isReadable  isSequential  isSignalConnected  isTextModeEnabled  isTransactionStarted  isWritable  open  openMode  peek  pos  putChar | read  readAll  readChannelCount  readChannelFinished  readData  readLine  readLineData  readyRead  receivers  reset  rollbackTransaction  seek  sender  senderSignalIndex  setCurrentReadChannel | setCurrentWriteChannel  setErrorString  setOpenMode  setTextModeEnabled  size  skip  startTransaction  timerEvent  ungetChar  waitForBytesWritten  waitForReadyRead  write  writeChannelCount  writeData |

【详细描述】用于支持读写数据块的device提供通用的接口；

atEnd：用于判断如果你到了文件的末尾

readLine：读取一行的数据；

readAll：读取文件所有数据，返回QByteArray对象；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

QFileDevice

【继承关系】QtCore.QFileDevice -> QtCore.QIODevice –> QtCore.QObject/QtCore.QIODeviceBase

【继承者】QSaveFile, QFile, QTemporaryFile

|  |  |  |  |
| --- | --- | --- | --- |
| atEnd  close  error  fileName  fileTime | flush  handle  isSequential  map  permissions | pos  resize  seek  setFileTime  setPermissions | size  unmap  unsetError |

【详细描述】

fileName：返回文件名；

fileTime：返回文件的FileTime相关信息，参数QFileDevice.FileTime对象[1. QFileDevice.FileAccessTime，最近访问时间；2. QFileDevice.FileBirthTime，创建事件；3. QFileDevice.FileMetadataChangeTime，上次修改元数据时间；4. QFileDevice.FileModificationTime，最近修改时间]

permissions：返回文件的权限；setPermissions：设置文件的权限，参数为QFileDevice.Permission对象[1. QFileDevice.ReadOwner；2. QFileDevice.WriteOwner；3. QFileDevice.ExeOwner；4. QFileDevice.ReadUser；5. QFileDevice.WriteUser；6. QFileDevice.ExeUser；7. QFileDevice.ReadGroup；8. QFileDevice.WriteGroup；9. QFileDevice.ExeGroup；10. QFileDevice.ReadOther；11. QFileDevice.WriteOther；12. QFileDevice.ExeOther]

QFileDevice.FileError对象：1.QFileDevice.NoError，无错误；2.QFileDevice.ReadError，读取文件错误；3.QFileDevice.WriteError，写入文件错误；4.QFileDevice.FatalError，之名错误；5.QFileDevice.ResourceError，资源错误；6.QFileDevice.OpenError，无法打开；7.QFileDevice.AbortError，禁止操作；8.QFileDevice.TimeOutError，超时；9.QFileDevice.UnspecifiedError，未指定的错误；10.QFileDevice.RemoveError，无法删除文件；11.QFileDevice.RenameError，无法重新命名；12.QFileDevice.PositionError，无法更改位置；13.QFileDevice.ResizeError，无法更改大小；14.QFileDevice.PermissionsError，权限问题无法访问；15.QFileDevice.CopyError，无法复制；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

## QFile

【继承关系】QtCore.QFile -> QtCore.QFileDevice -> QtCore.QIODevice –> QtCore.QObject/QtCore.QIODeviceBase

【继承者】QTemporaryFile

|  |  |  |  |
| --- | --- | --- | --- |
| childEvent  connectNotify  copy  customEvent  decodeName  disconnectNotify  encodeName  exists | fileName  isSignalConnected  link  moveToTrash  open  permissions  readData  readLineData | receivers  remove  rename  resize  sender  senderSignalIndex  setErrorString  setFileName | setOpenMode  setPermissions  size  symLinkTarget  timerEvent  writeData |

【详细描述】QFile类提供一个用于读/写文件的接口，它是一个可以用来读写文本文件、二进制文件和Qt资源的I/O 设备；

open：以OpenMode方式打开文件，参数为QIODevice.OpenMode对象[1. QIODevice.NotOpen，设备没有打开；2. QIODevice.ReadOnly，以只读方式打开,无法写入；3. QIODevice.WriteOnly，以只写方式打开，无法读取；4. QIODevice.Append，以附加模式打开，所有的数据都将写到文件末尾；5. QIODevice.Truncale，重写设备，如果可能设备在打开时会被截断，数据都将会丢失；6. QIODevice.Text在读取时，将行结束符转换成\n，在写入时，将结束符转换成本地格式，win32平台是\r\n；7. QIODevice.Unbuffered，绕过设备所有的缓冲区；

#--------------------------------------------------------------------------------------------------------------------------------------------------------------#

# PyQt5其他技巧

## QSS的用法：

选择器{属性：值}，选择器可以联和使用，采用逗号隔开，使用SetStyleSheet进行设置；

QSS选择器有1. 通配选择器：\*；2. 类型选择器：QPushButton，匹配所有的QPushButton类及其子类的实例；3. 属行选择器：QPushButton[name=”myBtn”]；4. 类选择器：.QPushButton，匹配所有的QPushButton实例，但是并不匹配其子类；5.objectName选择器，#objectName，匹配所有以objectName命名的控件；6. 后代选择器，QDialog QPushButton，匹配所有的QDialog容器中包含的QPushButton，不管是直接的还是间接的；7. 子选择器：QDialog > QPushButton，匹配所有的QDialog容器中包含的QPushButton，要求QPushButton直接父容器为QDialog；8. 子控件选择器：：，QComboBox::drop-down指下拉箭头；9. 伪状态选择器：

## Qt Designer用法

Preview (Ctrl + R)：预览

Widget Editing Mode（F3）: 用来设计用户界面；

Signals and Slots Editing Mode（F4）: 用拖曳的方式连接Signals和Slots

Buddy Editing Mode：用拖曳的方式连接标签和其他对象；

Tab Order Editing Mode：编辑Tab顺序。双击一个对象为1，然后依次点击剩余对象来设置顺序。

Selecting Objects,