



Week 12

Pemrograman Visual





Outline

- ❏ Assignment Review
- ❏ QStatusBar Widget
- ❏ QList Widget
- ❏ QPixmap Class

Assignment Review

QStatusBar Widget

QMainWindow object reserves a horizontal bar at the bottom as the status bar. It is used to display either permanent or contextual status information.

There are three types of status indicators:

- Temporary – Briefly occupies most of the status bar. For example, used to explain tool tip texts or menu entries.
- Normal – Occupies part of the status bar and may be hidden by temporary messages. For example, used to display the page and line number in a word processor.
- Permanent – It is never hidden. Used for important mode indications. For example, some applications put a Caps Lock indicator in the status bar.

Status bar of QMainWindow is retrieved by statusBar() function. setStatusBar() function activates it.

```
self.statusBar= QStatusBar()  
self.setStatusBar(self.statusBar)
```

| | |
|----------------------|--|
| addWidget() | Adds the given widget object in the status bar |
| addPermanentWidget() | Adds the given widget object in the status bar permanently |
| showMessage() | Displays a temporary message in the status bar for a specified time interval |
| clearMessage() | Removes any temporary message being shown |
| removeWidget() | Removes specified widget from the status bar |

Example



QList Widget

QListWidget class is an item-based interface to add or remove items from a list. Each item in the list is a QListWidgetItem object. QListWidget can be set to be multiselectable.

Following are the frequently used methods of QListWidget class:

| | |
|------------------|---|
| addItem() | Adds QListWidgetItem object or string in the list |
| addItems() | Adds each item in the list |
| insertItem() | Inserts item at the specified index |
| clear() | Removes contents of the list |
| setCurrentItem() | Sets currently selected item programmatically |
| sortItems() | Rearranges items in ascending order |

Following are the signals emitted by QListWidget:

| | |
|-----------------------------------|---|
| <code>currentItemChanged()</code> | Whenever current item changes |
| <code>itemClicked()</code> | Whenever an item in the list is clicked |

Example

QPixmap Class

QPixmap class provides an off-screen representation of an image. It can be used as a QPaintDevice object or can be loaded into another widget, typically a label or button.

Qt API has another similar class QImage, which is optimized for I/O and other pixel manipulations. QPixmap, on the other hand, is optimized for showing it on screen. Both formats are interconvertible.

The types of image files that can be read into a QPixmap object are as follows:

| | |
|------|---------------------------------------|
| BMP | Windows Bitmap |
| GIF | Graphic Interchange Format (optional) |
| JPG | Joint Photographic Experts Group |
| JPEG | Joint Photographic Experts Group |
| PNG | Portable Network Graphics |
| PBM | Portable Bitmap |
| PGM | Portable Graymap |
| PPM | Portable Pixmap |
| XBM | X11 Bitmap |
| XPM | X11 Pixmap |

Following methods are useful in handling QPixmap object:

| | |
|--------------|--|
| copy() | Copies pixmap data from a QRect object |
| fromImage() | Converts QImage object into QPixmap |
| grabWidget() | Creates a pixmap from the given widget |
| grabWindow() | Create pixmap of data in a window |
| Load() | Loads an image file as pixmap |
| save() | Saves the QPixmap object as a file |
| toImage | Converts a QPixmap to QImage |

Example





Thanks