# QTextEdit

The QTextEdit widget is an advanced editor that supports formatted rich text. It can be used to display HTML and other rich document formats. Internally, QTextEdit uses the QTextDocument class to describe both the high-level structure of each document and the low-level formatting of paragraphs.

If you are viewing this document in the *textedit* demo, you can edit this document to explore Qt's rich text editing features. We have included some comments in each of the following sections to encourage you to experiment.

## Font and Paragraph Styles

QTextEdit supports **bold**, *italic*, and <u>underlined</u> font styles, and can display **multicolored** <span style="color:red">text</span>. Font families such as Times New Roman and `Courier` can also be used directly. *If you place the cursor in a region of styled text, the controls in the tool bars will change to reflect the current style.*

Paragraphs can be formatted so that the text is left-aligned, right-aligned, centered, or fully justified.

*Try changing the alignment of some text and resize the editor to see how the text layout changes.*

## Lists

Different kinds of lists can be included in rich text documents. Standard bullet lists can be nested, using different symbols for each level of the list:

- Disc symbols are typically used for top-level list items.
  - Circle symbols can be used to distinguish between items in lower-level lists.
    - Square symbols provide a reasonable alternative to discs and circles.
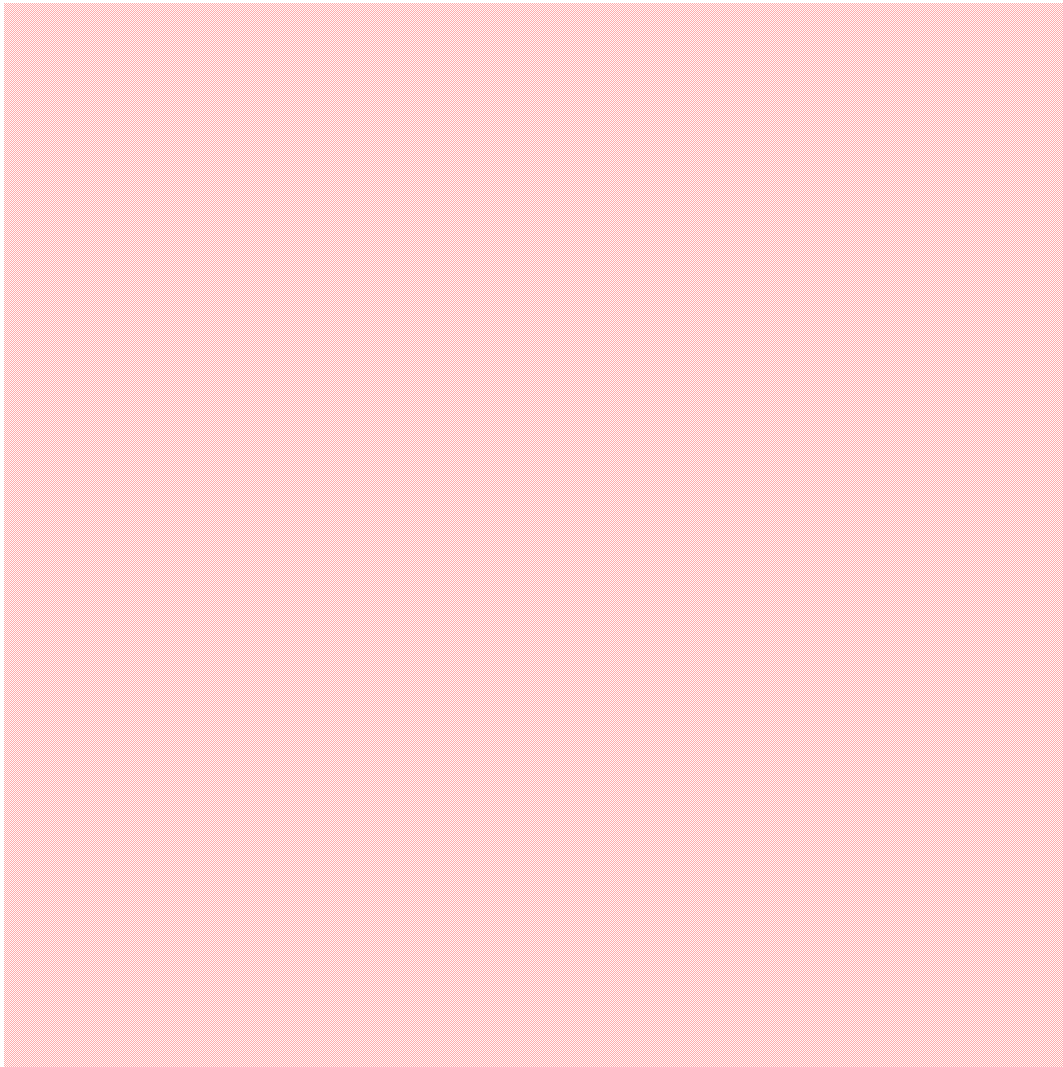
Ordered lists can be created that can be used for tables of contents. Different characters can be used to enumerate items, and we can use both Roman and Arabic numerals in the same list structure:

1. Introduction
2. Qt Tools
   a. Qt Assistant
   b. Qt Designer
      A. Form Editor
      B. Component Architecture
   c. Qt Linguist

The list will automatically be renumbered if you add or remove items. *Try adding new sections to the above list or removing existing item to see the numbers change.*

# Images

Inline images are treated like ordinary ranges of characters in the text editor, so they flow with the surrounding text. Images can also be selected in the same way as text, making it easy to cut, copy, and paste them.

*Try to select this image by clicking and dragging over it with the mouse, or use the text cursor to select it by holding down Shift and using the arrow keys. You can then cut or copy it, and paste it into different parts of this document.*

# Tables

QTextEdit can arrange and format tables, supporting features such as row and column spans, text formatting within cells, and size constraints for columns.

|               | Development Tools | Programming Techniques | Graphical User Interfaces |
|---------------|-------------------|------------------------|---------------------------|
| **9:00 – 11:00** | Introduction to *Qt* | | |
| **11:00 – 13:00** | Using *qmake* | Object–oriented Programming | Layouts in *Qt* |

| 13:00 – 15:00 | Qt Designer Tutorial | Extreme Programming | Writing Custom Styles |
|---|---|---|---|
| 15:00 – 17:00 | Qt Linguist and Internationalization | | |

*Try adding text to the cells in the table and experiment with the alignment of the paragraphs.*

# Hyperlinks

QTextEdit is designed to support hyperlinks between documents, and this feature is used extensively in *Qt Assistant.* Hyperlinks are automatically created when an HTML file is imported into an editor. Since the rich text framework supports hyperlinks natively, they can also be created programatically.

# Undo and Redo

Full support for undo and redo operations is built into QTextEdit and the underlying rich text framework. Operations on a document can be packaged together to make editing a more comfortable experience for the user.

*Try making changes to this document and press Ctrl+Z to undo them. You can always recover  the original contents of the document.*