QTextBrowser Class

QTextBrowser 类提供了具有超文本导航功能的富文本浏览器。更多的...

Header:	**#include**
CMake:	find_package(Qt6 REQUIRED COMPONENTS Widgets) target_link_libraries(mytarget PRIVATE Qt6::Widgets)
qmake:	QT += widgets
Inherits:	QTextEdit

- 所有成员的列表，包括继承的成员
- QTextBrowser 是富文本处理 API的一部分。

特性

modified : const boolopenExternalLinks : boolopenLinks : boolreadOnly : const bool	**searchPaths : QStringListsource : QUrlsourceType : const QTextDocument::ResourceTypeundoRedoEnabled : const bool**

公共函数

	QTextBrowser(QWidget **parent* = nullptr)
int	**backwardHistoryCount**() const
void	**clearHistory**()
int	**forwardHistoryCount**() const
QString	**historyTitle**(int *i*) const
QUrl	**historyUrl**(int *i*) const
bool	**isBackwardAvailable**() const
bool	**isForwardAvailable**() const
bool	**openExternalLinks**() const
bool	**openLinks**() const

QStringList	searchPaths() const
	QTextBrowser(QWidget *parent = nullptr)
void	setOpenExternalLinks(bool *open*)
void	setOpenLinks(bool *open*)
void	setSearchPaths(const QStringList &*paths*)
QUrl	source() const
QTextDocument::ResourceType	sourceType() const

重载的函数

virtual QVariant	**loadResource(int *type*, const QUrl &*name*) override**

公共槽

virtual void	**backward()**
virtual void	forward()
virtual void	home()
virtual void	reload()
void	setSource(const QUrl &*url*, QTextDocument::ResourceType *type* = QTextDocument::UnknownResource)

信号

void	**anchorClicked(const QUrl &*link*)**
void	backwardAvailable(bool *available*)
void	forwardAvailable(bool *available*)
void	highlighted(const QUrl &*link*)
void	historyChanged()
void	sourceChanged(const QUrl &*src*)

protected function

virtual void	**doSetSource(const QUrl &*url*, QTextDocument::ResourceType *type* = QTextDocument::UnknownResource)**

重载的protected function

virtual bool	**event(QEvent **e*) override**
virtual bool	**focusNextPrevChild**(bool *next*) override
virtual void	**focusOutEvent**(QFocusEvent **ev*) override
virtual void	**keyPressEvent**(QKeyEvent **ev*) override
virtual void	**mouseMoveEvent**(QMouseEvent **e*) override
virtual void	**mousePressEvent**(QMouseEvent **e*) override
virtual void	**mouseReleaseEvent**(QMouseEvent **e*) override
virtual void	**paintEvent**(QPaintEvent **e*) override

详细说明

这个类扩展了QTextEdit（在只读模式下），添加一些导航功能，以便用户可以跟踪超文本文档中的链接。

如果您想为用户提供可编辑的富文本编辑器，请使用QTextEdit。如果您想要一个没有超文本导航的文本浏览器，请使用QTextEdit，并使用QTextEdit::setReadOnly() 禁用编辑。如果您只需要显示一小段富文本使用QLabel。

文档来源及内容

的内容QTextEdit设置为setHtml （） 或者setPlainText()，但 QTextBrowser 也实现了setSource() 函数，使得可以使用命名文档作为源文本。在搜索路径列表和当前文档工厂的目录中查找该名称。

如果文档名称以锚点结尾（例如，"`#anchor`"），则文本浏览器会自动滚动到该位置（使用scrollToAnchor()）。当用户点击超链接时，浏览器将调用setSource() 本身以链接的`href`值作为参数。您可以通过连接到来跟踪当前源sourceChanged （） 信号。

导航

QTextBrowser提供backward （） 和forward() 插槽可用于实现后退和前进按钮。这home() 槽将文本设置为显示的第一个文档。这anchorClicked当用户单击锚点时，会发出 () 信号。要覆盖浏览器的默认导航行为，请调用setSource() 函数在连接到该信号的槽中提供新的文档文本。

如果要加载存储在 Qt 资源系统中的文档，请使用qrcURL 中的方案来加载。例如，对于文档资源路径，:/docs/index.html请使用qrc:/docs/index.html以下 URL 作为 URL：setSource()。

也可以看看QTextEdit和QTextDocument。

属性文档

[read-only]modified : const bool

该属性保存文本浏览器的内容是否被修改

openExternalLinks : bool

指定是否QTextBrowser应该使用自动打开到外部源的链接QDesktopServices::openUrl() 而不是发出anchorClicked信号。如果链接的方案既不是 file 也不是 qrc，则链接被视为外部链接。

默认值为 false。

访问功能：

bool	**openExternalLinks() const**
void	**setOpenExternalLinks**(bool *open*)

openLinks : bool

该属性指定是否QTextBrowser应自动打开用户尝试通过鼠标或键盘激活的链接。

无论该房产的价值如何anchorClicked信号始终发出。

默认值是true。

访问功能：

bool	**openLinks() const**
void	**setOpenLinks**(bool *open*)

readOnly : const bool

该属性保存文本浏览器是否是只读的

默认情况下，该属性为true。

searchPaths : QStringList

该属性保存文本浏览器用来查找支持内容的搜索路径

QTextBrowser使用此列表来查找图像和文档。

默认情况下，此属性包含一个空字符串列表。

访问功能：

QStringList	searchPaths() const
void	**setSearchPaths**(const QStringList &*paths*)

source : QUrl

该属性保存显示文档的名称。

如果未显示任何文档或来源未知，则该 URL 无效。

设置该属性时QTextBrowser尝试在路径中查找具有指定名称的文档searchPaths当前源的属性和目录，除非该值是绝对文件路径。它还检查可选锚点并相应地滚动文档

如果文档中的第一个标签是`<qt type=detail>`，则该文档将显示为弹出窗口，而不是浏览器窗口本身中的新文档。否则，文档将在文本浏览器中正常显示，其中文本设置为指定文档的内容QTextDocument::setHtml （） 或者QTextDocument::setMarkdown()，具体取决于文件名是否以任何已知的 Markdown 文件扩展名结尾。

如果您想避免自动类型检测并显式指定类型，请调用setSource() 而不是设置此属性。

默认情况下，此属性包含一个空 URL。

访问功能：

QUrl	source() const
void	**setSource**(const QUrl &*url*, QTextDocument::ResourceType *type* = QTextDocument::UnknownResource)

[read-only]sourceType : const QTextDocument::ResourceType

该属性保存显示文档的类型

这是QTextDocument::UnknownResource如果没有显示文档或者源类型未知。否则，它保存检测到的类型，或指定的类型setSource （） 被称为。

访问功能：

QTextDocument::ResourceType	sourceType() const

undoRedoEnabled : const bool

该属性保存文本浏览器是否支持撤消/重做操作

默认情况下，该属性为`false`。

成员函数文档

*[explicit]QTextBrowser::QTextBrowser(QWidget *parent = nullptr)*

构造一个空的 QTextBrowser 及其父级*parent*。

[signal]void QTextBrowser::anchorClicked(const QUrl &link)

当用户单击锚点时会发出此信号。传入锚点引用的 URL*link*。

请注意，浏览器将自动处理导航到由*link*除非openLinks属性设置为 false 或者您调用setSource() 在一个已连接的槽中。该机制用于覆盖浏览器的默认导航功能。

[virtual slot]void QTextBrowser::backward()

将显示的文档更改为通过导航链接构建的文档列表中的上一个文档。如果没有先前的文档，则不执行任何操作。

也可以看看forward（） 和backwardAvailable()。

[signal]void QTextBrowser::backwardAvailable(bool available)

当可用时会发出此信号backward（） 变化。*available*当用户位于home(); 否则就是真的。

int QTextBrowser::backwardHistoryCount() const

返回历史记录中向后的位置数。

void QTextBrowser::clearHistory()

清除访问过的文档的历史记录并禁用向前和向后导航。

也可以看看backward（） 和forward()。

[virtual protected]void QTextBrowser::doSetSource(const QUrl &url, QTextDocument::ResourceType type = QTextDocument::UnknownResource)

尝试在给定的位置加载文档*url*与指定的*type*。

setSource() 调用 doSetSource。在 Qt 5 中，setSource（常量QUrl&url) 是虚拟的。在 Qt 6 中，doSetSource() 是虚拟的，因此可以在子类中重写它。

*[override virtual protected]bool QTextBrowser::event(QEvent *e)*

重新实现：QAbstractScrollArea::event（QEvent *事件）。

[override virtual protected]bool QTextBrowser::focusNextPrevChild(bool next)

重新实现：QTextEdit::focusNextPrevChild（接下来是布尔值）。

*[override virtual protected]void QTextBrowser::focusOutEvent(QFocusEvent *ev)*

重新实现：QTextEdit::focusOutEvent（QFocusEvent *e）。

[virtual slot]void QTextBrowser::forward()

将显示的文档更改为通过导航链接构建的文档列表中的下一个文档。如果没有下一个文档，则不执行任何操作。

也可以看看backward（） 和forwardAvailable()。

[signal]void QTextBrowser::forwardAvailable(bool available)

当可用时会发出此信号forward（） 变化。*available*用户导航后为 truebackward当用户导航或离开时 () 和 falseforward()。

int QTextBrowser::forwardHistoryCount() const

返回历史记录中前进的位置数。

[signal]void QTextBrowser::highlighted(const QUrl &link)

当用户选择但未激活文档中的锚点时，会发出此信号。传入锚点引用的 URL*link*。

[signal]void QTextBrowser::historyChanged()

当历史记录发生变化时，会发出此信号。

也可以看看historyTitle（） 和historyUrl()。

QString QTextBrowser::historyTitle(int i) const

返回documentTitleHistoryItem 的 ()。

输入	返回
i< 0	backward（） 历史
i== 0	当前，参见QTextBrowser::source()
i> 0	forward（） 历史

```
backaction.setToolTip(browser.historyTitle(-1));
forwardaction.setToolTip(browser.historyTitle(+1));
```

QUrl QTextBrowser::historyUrl(int i) const

返回 HistoryItem 的 url。

输入	返回
i< 0	backward（） 历史

输入	返回
$i == 0$	当前，参见QTextBrowser::source()
$i > 0$	forward（） 历史

[virtual slot]void QTextBrowser::home()

将显示的文档更改为历史记录中的第一个文档。

bool QTextBrowser::isBackwardAvailable() const

返回true文本浏览器是否可以使用以下命令在文档历史记录中后退backward()。

也可以看看backwardAvailable（） 和backward()。

bool QTextBrowser::isForwardAvailable() const

返回true文本浏览器是否可以使用以下命令在文档历史记录中前进forward()。

也可以看看forwardAvailable（） 和forward()。

*[override virtual protected]void QTextBrowser::keyPressEvent(QKeyEvent *ev)*

重新实现：QTextEdit::keyPressEvent（QKeyEvent *e）。

事件*ev*用于提供以下键盘快捷键：

按键	行动
Alt+向左箭头	backward()
Alt+右箭头	forward()
Alt+向上键	home()

[override virtual]QVariant QTextBrowser::loadResource(int type, const QUrl &name)

重新实现：QTextEdit::loadResource（int 类型，const QUrl 和名称）。

加载文档时以及针对文档中的每个图像调用此函数。这*type*表示要加载的资源类型。无效QVariant如果资源无法加载则返回。

默认实现忽略*type*并尝试通过解释来定位资源*name*作为文件名。如果它不是绝对路径，它会尝试在路径中查找该文件searchPaths属性并与当前源位于同一目录中。成功后，结果是QVariant存储一个QByteArray与文件的内容。

如果重新实现这个函数，可以返回其他的QVariant类型。下表显示了根据资源类型支持的变体类型：

资源类型	QMetaType::Type
QTextDocument::HtmlResource	QString或者QByteArray
QTextDocument::ImageResource	QImage,QPixmap或者QByteArray
QTextDocument::StyleSheetResource	QString或者QByteArray
QTextDocument::MarkdownResource	QString或者QByteArray

*[override virtual protected]void QTextBrowser::mouseMoveEvent(QMouseEvent *e)*

重新实现：QTextEdit::mouseMoveEvent(QMouseEvent *e)。

*[override virtual protected]void QTextBrowser::mousePressEvent(QMouseEvent *e)*

重新实现：QTextEdit::mousePressEvent(QMouseEvent *e)。

*[override virtual protected]void QTextBrowser::mouseReleaseEvent(QMouseEvent *e)*

重新实现：QTextEdit::mouseReleaseEvent(QMouseEvent *e)。

*[override virtual protected]void QTextBrowser::paintEvent(QPaintEvent *e)*

重新实现：QTextEdit::paintEvent（QPaintEvent *事件）。

[virtual slot]void QTextBrowser::reload()

重新加载当前设置的源。

[slot]void QTextBrowser::setSource(const QUrl &url, QTextDocument::ResourceType type = QTextDocument::UnknownResource)

尝试在给定的位置加载文档*url*与指定的*type*。

如果*type*是UnknownResource（默认），将检测文档类型：即，如果 url 以 ,.md或.mkd扩展名结尾.markdown，则将通过以下方式加载文档QTextDocument::setMarkdown(); 否则它将通过加载QTextDocument::setHtml()。可以通过指定绕过此检测*type*明确地。

注意：属性的 Setter 函数source。

也可以看看source()。

[signal]void QTextBrowser::sourceChanged(const QUrl &src)

当源发生变化时会发出此信号，*src*成为新的来源。

调用时会以编程方式发生源更改setSource(),forward(),backward（） 或者home() 或当用户单击链接或按下等效的按键序列时。