



# QListWidgetItem Class Reference

## [QtGui module]

The QListWidgetItem class provides an item for use with the [QListWidget](#) item view class. [More...](#)

### Types

- enum [ItemType](#) { Type, UserType }

### Methods

- [\\_\\_init\\_\\_](#) (*self*, [QListWidget](#) *parent* = None, int *type* = [QListWidgetItem.Type](#))
- [\\_\\_init\\_\\_](#) (*self*, [QString](#) *text*, [QListWidget](#) *parent* = None, int *type* = [QListWidgetItem.Type](#))
- [\\_\\_init\\_\\_](#) (*self*, [QIcon](#) *icon*, [QString](#) *text*, [QListWidget](#) *parent* = None, int *type* = [QListWidgetItem.Type](#))
- [\\_\\_init\\_\\_](#) (*self*, [QListWidgetItem](#) *other*)
- [QBrush](#) [background](#) (*self*)
- [QColor](#) [backgroundColor](#) (*self*)
- Qt.CheckState [checkState](#) (*self*)
- [QListWidgetItem](#) [clone](#) (*self*)
- [QVariant](#) [data](#) (*self*, int *role*)
- Qt.ItemFlags [flags](#) (*self*)
- [QFont](#) [font](#) (*self*)
- [QBrush](#) [foreground](#) (*self*)
- [QIcon](#) [icon](#) (*self*)
- bool [isHidden](#) (*self*)
- bool [isSelected](#) (*self*)
- [QListWidget](#) [listWidget](#) (*self*)
- [read](#) (*self*, [QDataStream](#) *in*)
- [setBackground](#) (*self*, [QBrush](#) *brush*)
- [setBackgroundColor](#) (*self*, [QColor](#) *color*)
- [setCheckState](#) (*self*, Qt.CheckState *state*)
- [setData](#) (*self*, int *role*, [QVariant](#) *value*)
- [setFlags](#) (*self*, Qt.ItemFlags *aflags*)
- [setFont](#) (*self*, [QFont](#) *afont*)
- [setForeground](#) (*self*, [QBrush](#) *brush*)
- [setHidden](#) (*self*, bool *ahide*)
- [setIcon](#) (*self*, [QIcon](#) *aicon*)
- [setSelected](#) (*self*, bool *aselect*)
- [setSizeHint](#) (*self*, [QSize](#) *size*)
- [setStatusTip](#) (*self*, [QString](#) *astatusTip*)
- [setText](#) (*self*, [QString](#) *atext*)
- [setTextAlignment](#) (*self*, int *alignment*)
- [setTextColor](#) (*self*, [QColor](#) *color*)
- [setToolTip](#) (*self*, [QString](#) *atoolTip*)
- [setWhatsThis](#) (*self*, [QString](#) *awhatsThis*)
- [QSize](#) [sizeHint](#) (*self*)
- [QString](#) [statusTip](#) (*self*)
- [QString](#) [text](#) (*self*)

- int `textAlignment` (*self*)
- QColor `textColor` (*self*)
- QString `toolTip` (*self*)
- int `type` (*self*)
- QString `whatsThis` (*self*)
- `write` (*self*, QDataStream *out*)

## Special Methods

- bool `__ge__` (*self*, QListWidgetItem *other*)
  - bool `__lt__` (*self*, QListWidgetItem *other*)
- 

## Detailed Description

The `QListWidgetItem` class provides an item for use with the `QListWidget` item view class.

A `QListWidgetItem` represents a single item in a `QListWidget`. Each item can hold several pieces of information, and will display them appropriately.

The item view convenience classes use a classic item-based interface rather than a pure model/view approach. For a more flexible list view widget, consider using the `QListView` class with a standard model.

List items can be inserted automatically into a list, when they are constructed, by specifying the list widget:

```
new QListWidgetItem(tr("Hazel"), listWidget);
```

Alternatively, list items can also be created without a parent widget, and later inserted into a list using `QListWidget.insertItem()`.

List items are typically used to display `text()` and an `icon()`. These are set with the `setText()` and `setIcon()` functions. The appearance of the text can be customized with `setFont()`, `setForeground()`, and `setBackground()`. Text in list items can be aligned using the `setTextAlignment()` function. Tooltips, status tips and "What's This?" help can be added to list items with `setToolTip()`, `setStatusTip()`, and `setWhatsThis()`.

By default, items are enabled, selectable, checkable, and can be the source of drag and drop operations.

Each item's flags can be changed by calling `setFlags()` with the appropriate value (see [Qt.ItemFlags](#)). Checkable items can be checked, unchecked and partially checked with the `setCheckState()` function. The corresponding `checkState()` function indicates the item's current check state.

The `isHidden()` function can be used to determine whether the item is hidden. To hide an item, use `setHidden()`.

## Subclassing

When subclassing `QListWidgetItem` to provide custom items, it is possible to define new types for them enabling them to be distinguished from standard items. For subclasses that require this feature, ensure that you call the base class constructor with a new type value equal to or greater than `UserType`, within *your* constructor.

---

# Type Documentation

## QListWidgetItem.ItemType

This enum describes the types that are used to describe list widget items.

Constant	Value	Description
<code>QListWidgetItem.Type</code>	0	The default type for list widget items.
<code>QListWidgetItem.UserType</code>	1000	The minimum value for custom types. Values below <code>UserType</code> are reserved by Qt.

You can define new user types in `QListWidgetItem` subclasses to ensure that custom items are treated specially.

See also `type()`.

## Method Documentation

```
QListWidgetItem.__init__(self, QListWidget parent = None,
                          int type = QListWidgetItem.Type)
```

The *parent* argument, if not `None`, causes *self* to be owned by Qt instead of PyQt.

Constructs an empty list widget item of the specified *type* with the given *parent*. If *parent* is not specified, the item will need to be inserted into a list widget with `QListWidget.insertItem()`.

This constructor inserts the item into the model of the parent that is passed to the constructor. If the model is sorted then the behavior of the insert is undetermined since the model will call the '`<`' operator method on the item which, at this point, is not yet constructed. To avoid the undetermined behavior, we recommend not to specify the parent and use `QListWidget.insertItem()` instead.

See also `type()`.

```
QListWidgetItem.__init__(self, QString text,
                          QListWidget parent = None, int type = QListWidgetItem.Type)
```

The *parent* argument, if not `None`, causes *self* to be owned by Qt instead of PyQt.

Constructs an empty list widget item of the specified *type* with the given *text* and *parent*. If the parent is not specified, the item will need to be inserted into a list widget with `QListWidget.insertItem()`.

This constructor inserts the item into the model of the parent that is passed to the constructor. If the model is sorted then the behavior of the insert is undetermined since the model will call the '`<`' operator method on the item which, at this point, is not yet constructed. To avoid the undetermined behavior, we recommend not to specify the parent and use `QListWidget.insertItem()` instead.

See also `type()`.

```
QListWidgetItem.__init__(self, QIcon icon, QString text,
                          QListWidget parent = None, int type = QListWidgetItem.Type)
```

The *parent* argument, if not None, causes *self* to be owned by Qt instead of PyQt.

Constructs an empty list widget item of the specified *type* with the given *icon*, *text* and *parent*. If the parent is not specified, the item will need to be inserted into a list widget with [QListWidget.insertItem\(\)](#).

This constructor inserts the item into the model of the parent that is passed to the constructor. If the model is sorted then the behavior of the insert is undetermined since the model will call the '`<`' operator method on the item which, at this point, is not yet constructed. To avoid the undetermined behavior, we recommend not to specify the parent and use [QListWidget.insertItem\(\)](#) instead.

See also [type\(\)](#).

**QListWidgetItem.\_\_init\_\_** (*self*, [QListWidgetItem](#) *other*)

Constructs a copy of *other*. Note that [type\(\)](#) and [listWidget\(\)](#) are not copied.

This function is useful when reimplementing [clone\(\)](#).

This function was introduced in Qt 4.1.

See also [data\(\)](#) and [flags\(\)](#).

**QBrush** [QListWidgetItem.background](#) (*self*)

Returns the brush used to display the list item's background.

This function was introduced in Qt 4.2.

See also [setBackground\(\)](#) and [foreground\(\)](#).

**QColor** [QListWidgetItem.backgroundColor](#) (*self*)

**Qt.CheckState** [QListWidgetItem.checkState](#) (*self*)

Returns the checked state of the list item (see [Qt.CheckState](#)).

See also [setCheckState\(\)](#) and [flags\(\)](#).

**QListWidgetItem** [QListWidgetItem.clone](#) (*self*)

Creates an exact copy of the item.

**QVariant** [QListWidgetItem.data](#) (*self*, int *role*)

Returns the item's data for a given *role*. Reimplement this function if you need extra roles or special behavior for certain roles.

See also [Qt.ItemDataRole](#) and [setData\(\)](#).

**Qt.ItemFlags** [QListWidgetItem.flags](#) (*self*)

Returns the item flags for this item (see [Qt.ItemFlags](#)).

See also [setFlags\(\)](#).

### `QFont` `QListWidgetItem.font (self)`

Returns the font used to display this list item's text.

See also `setFont()`.

### `QBrush` `QListWidgetItem.foreground (self)`

Returns the brush used to display the list item's foreground (e.g. text).

This function was introduced in Qt 4.2.

See also `setForeground()` and `background()`.

### `QIcon` `QListWidgetItem.icon (self)`

Returns the list item's icon.

See also `setIcon()` and `iconSize`.

### `bool` `QListWidgetItem.isHidden (self)`

Returns true if the item is hidden; otherwise returns false.

This function was introduced in Qt 4.2.

See also `setHidden()`.

### `bool` `QListWidgetItem.isSelected (self)`

Returns true if the item is selected; otherwise returns false.

This function was introduced in Qt 4.2.

See also `setSelected()`.

### `QListWidget` `QListWidgetItem.listWidget (self)`

Returns the list widget containing the item.

### `QListWidgetItem.read (self, QDataStream in)`

Reads the item from stream *in*.

See also `write()`.

### `QListWidgetItem.setBackground (self, QBrush brush)`

Sets the background brush of the list item to the given *brush*.

This function was introduced in Qt 4.2.

See also `background()` and `setForeground()`.

### `QListWidgetItem.setBackgroundColor (self, QColor color)`

**QListWidgetItem.setCheckState** (*self*, [Qt.CheckState](#) *state*)

Sets the check state of the list item to *state*.

See also [checkState\(\)](#).

**QListWidgetItem.setData** (*self*, int *role*, QVariant *value*)

Sets the data for a given *role* to the given *value*. Reimplement this function if you need extra roles or special behavior for certain roles.

See also [Qt.ItemDataRole](#) and [data\(\)](#).

**QListWidgetItem.setFlags** (*self*, [Qt.ItemFlags](#) *aflags*)

Sets the item flags for the list item to *flags*.

See also [flags\(\)](#) and [Qt.ItemFlags](#).

**QListWidgetItem.setFont** (*self*, [QFont](#) *afont*)

Sets the font used when painting the item to the given *font*.

See also [font\(\)](#).

**QListWidgetItem.setForeground** (*self*, [QBrush](#) *brush*)

Sets the foreground brush of the list item to the given *brush*.

This function was introduced in Qt 4.2.

See also [foreground\(\)](#) and [setBackground\(\)](#).

**QListWidgetItem.setHidden** (*self*, bool *ahide*)

Hides the item if *hide* is true; otherwise shows the item.

This function was introduced in Qt 4.2.

See also [isHidden\(\)](#).

**QListWidgetItem.setIcon** (*self*, [QIcon](#) *aicon*)

Sets the icon for the list item to the given *icon*.

See also [icon\(\)](#), [text\(\)](#), and [iconSize](#).

**QListWidgetItem.setSelected** (*self*, bool *aselect*)

Sets the selected state of the item to *select*.

This function was introduced in Qt 4.2.

See also [isSelected\(\)](#).

**QListWidgetItem.setSizeHint** (*self*, [QSize](#) *size*)

Sets the size hint for the list item to be *size*. If no size hint is set, the item delegate will compute the size hint based on the item data.

This function was introduced in Qt 4.1.

See also [sizeHint\(\)](#).

**QListWidgetItem.setStatusTip** (*self*, QString *astatusTip*)

Sets the status tip for the list item to the text specified by *statusTip*. [QListWidgetItem](#) mouseTracking needs to be enabled for this feature to work.

See also [statusTip\(\)](#), [setToolTip\(\)](#), [setWhatsThis\(\)](#), and [QWidget.setMouseTracking\(\)](#).

**QListWidgetItem.setText** (*self*, QString *atext*)

Sets the text for the list widget item's to the given *text*.

See also [text\(\)](#).

**QListWidgetItem.setTextAlignment** (*self*, int *alignment*)

Sets the list item's text alignment to *alignment*.

See also [textAlignment\(\)](#) and [Qt.AlignmentFlag](#).

**QListWidgetItem.setTextColor** (*self*, QColor *color*)

**QListWidgetItem.setToolTip** (*self*, QString *atoolTip*)

Sets the tooltip for the list item to the text specified by *toolTip*.

See also [toolTip\(\)](#), [setStatusTip\(\)](#), and [setWhatsThis\(\)](#).

**QListWidgetItem.setWhatsThis** (*self*, QString *awhatsThis*)

Sets the "What's This?" help for the list item to the text specified by *whatsThis*.

See also [whatsThis\(\)](#), [setStatusTip\(\)](#), and [setToolTip\(\)](#).

**QSize** [QListWidgetItem.sizeHint](#) (*self*)

Returns the size hint set for the list item.

This function was introduced in Qt 4.1.

See also [setSizeHint\(\)](#).

**QString** [QListWidgetItem.statusTip](#) (*self*)

Returns the list item's status tip.

See also [setStatusTip\(\)](#).

**QString** [QListWidgetItem.text](#) (*self*)

Returns the list item's text.

See also [setText\(\)](#).

`int QListWidgetItem.textAlignment (self)`

Returns the text alignment for the list item.

See also [setTextAlignment\(\)](#) and [Qt.AlignmentFlag](#).

[QColor](#) `QListWidgetItem.textColor (self)`

`QString QListWidgetItem.toolTip (self)`

Returns the list item's tooltip.

See also [setToolTip\(\)](#), [statusTip\(\)](#), and [whatsThis\(\)](#).

`int QListWidgetItem.type (self)`

Returns the type passed to the [QListWidgetItem](#) constructor.

`QString QListWidgetItem.whatsThis (self)`

Returns the list item's "What's This?" help text.

See also [setWhatsThis\(\)](#), [statusTip\(\)](#), and [toolTip\(\)](#).

`QListWidgetItem.write (self, QDataStream out)`

Writes the item to stream *out*.

See also [read\(\)](#).

`bool QListWidgetItem.__ge__ (self, QListWidgetItem other)`

`bool QListWidgetItem.__lt__ (self, QListWidgetItem other)`