## Selector Types

All the examples so far used the simplest type of selector, the Type Selector. Qt Style Sheets support all the [selectors defined in CSS2](http://www.w3.org/TR/REC-CSS2/selector.html" \l "q1). The table below summarizes the most useful types of selectors.

| Selector | Example | Explanation |
| --- | --- | --- |
| Universal Selector | \* | Matches all widgets. |
| Type Selector | QPushButton | Matches instances of [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) and of its subclasses. |
| Property Selector | QPushButton[flat="false"] | Matches instances of [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) that are not [flat](https://doc.qt.io/qt-5/qpushbutton.html" \l "flat-prop). You may use this selector to test for any Qt [property](https://doc.qt.io/qt-5/properties.html) that supports [QVariant::toString](https://doc.qt.io/qt-5/qvariant.html" \l "toString)() (see the [toString()](https://doc.qt.io/qt-5/qvariant.html" \l "toString) function documentation for details). In addition, the special class property is supported, for the name of the class.  This selector may also be used to test dynamic properties. For more information on customization using dynamic properties, refer to [Customizing Using Dynamic Properties](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-using-dynamic-properties).  Instead of =, you can also use ~= to test whether a Qt property of type [QStringList](https://doc.qt.io/qt-5/qstringlist.html) contains a given [QString](https://doc.qt.io/qt-5/qstring.html).  **Warning:** If the value of the Qt property changes after the style sheet has been set, it might be necessary to force a style sheet recomputation. One way to achieve this is to unset the style sheet and set it again. |
| Class Selector | .QPushButton | Matches instances of [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), but not of its subclasses.  This is equivalent to \*[class~="QPushButton"]. |
| ID Selector | QPushButton#okButton | Matches all [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) instances whose [object name](https://doc.qt.io/qt-5/qobject.html" \l "objectName-prop) is okButton. |
| Descendant Selector | QDialog QPushButton | Matches all instances of [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) that are descendants (children, grandchildren, etc.) of a [QDialog](https://doc.qt.io/qt-5/qdialog.html). |
| Child Selector | QDialog > QPushButton | Matches all instances of [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) that are direct children of a [QDialog](https://doc.qt.io/qt-5/qdialog.html). |

## Sub-Controls

For styling complex widgets, it is necessary to access subcontrols of the widget, such as the drop-down button of a [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) or the up and down arrows of a [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html). Selectors may contain *subcontrols* that make it possible to restrict the application of a rule to specific widget subcontrols. For example:

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html)::drop-down { image: url(dropdown.png) }

The above rule styles the drop-down button of all [QComboBox](https://doc.qt.io/qt-5/qcombobox.html)es. Although the double-colon (::) syntax is reminiscent of CSS3 Pseudo-Elements, Qt Sub-Controls differ conceptually from these and have different cascading semantics.

Sub-controls are always positioned with respect to another element - a reference element. This reference element could be the widget or another Sub-control. For example, the [::drop-down](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "drop-down-sub) of a [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) is placed, by default, in the top right corner of the Padding rectangle of the [QComboBox](https://doc.qt.io/qt-5/qcombobox.html). The [::drop-down](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "drop-down-sub) is placed, by default, in the Center of the Contents rectangle of the [::drop-down](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "drop-down-sub) Sub-control. See the [List of Stylable Widgets](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "list-of-stylable-widgets) below for the Sub-controls to use to style a widget and their default positions.

The origin rectangle to be used can be changed using the [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop) property. For example, if we want to place the drop-down in the margin rectangle of the [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) instead of the default Padding rectangle, we can specify:

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html) {

margin-right: 20px;

}

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html)::drop-down {

subcontrol-origin: margin;

}

The alignment of the drop-down within the Margin rectangle is changed using [subcontrol-position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-position-prop) property.

The [width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "width-prop) and [height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "height-prop) properties can be used to control the size of the Sub-control. Note that setting a [image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "image-prop) implicitly sets the size of a Sub-control.

The relative positioning scheme ([position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) : relative), allows the position of the Sub-Control to be offset from its initial position. For example, when the [QComboBox](https://doc.qt.io/qt-5/qcombobox.html)'s drop-down button is pressed, we might like the arrow inside to be offset to give a "pressed" effect. To achieve this, we can specify:

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html)::down-arrow {

image: url(down\_arrow.png);

}

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html)::down-arrow:pressed {

position: relative;

top: 1px; left: 1px;

}

The absolute positioning scheme ([position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) : absolute), allows the position and size of the Sub-control to be changed with respect to the reference element.

Once positioned, they are treated the same as widgets and can be styled using the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).

See the [List of Sub-Controls](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "list-of-sub-controls) below for a list of supported sub-controls, and [Customizing the QPushButton's Menu Indicator Sub-Control](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-the-qpushbutton-s-menu-indicator-sub-control) for a realistic example.

**Note:**With complex widgets such as [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) and [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html), if one property or sub-control is customized, **all** the other properties or sub-controls must be customized as well.

## Pseudo-States

Selectors may contain *pseudo-states* that denote that restrict the application of the rule based on the widget's state. Pseudo-states appear at the end of the selector, with a colon (:) in between. For example, the following rule applies when the mouse hovers over a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover { color: white }

Pseudo-states can be negated using the exclamation operator. For example, the following rule applies when the mouse does not hover over a [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html):

[QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html):!hover { color: red }

Pseudo-states can be chained, in which case a logical AND is implied. For example, the following rule applies to when the mouse hovers over a checked [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html):

[QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html):hover:checked { color: white }

Negated Pseudo-states may appear in Pseudo-state chains. For example, the following rule applies when the mouse hovers over a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) that is not pressed:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover:!pressed { color: blue; }

If needed, logical OR can be expressed using the comma operator:

[QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html):hover, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html):checked { color: white }

Pseudo-states can appear in combination with subcontrols. For example:

[QComboBox](https://doc.qt.io/qt-5/qcombobox.html)::drop-down:hover { image: url(dropdown\_bright.png) }

See the [List of Pseudo-States](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "list-of-pseudo-states) section below for the list of pseudo-states provided by Qt widgets.

## Conflict Resolution

Conflicts arise when several style rules specify the same properties with different values. Consider the following style sheet:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html)#okButton { color: gray }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { color: red }

Both rules match [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) instances called okButton and there is a conflict for the color property. To resolve this conflict, we must take into account the *specificity* of the selectors. In the above example, QPushButton#okButton is considered more specific than QPushButton, because it (usually) refers to a single object, not to all instances of a class.

Similarly, selectors with pseudo-states are more specific than ones that do not specify pseudo-states. Thus, the following style sheet specifies that a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) should have white text when the mouse is hovering over it, otherwise red text:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover { color: white }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { color: red }

Here's a tricky one:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover { color: white }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):enabled { color: red }

Here, both selectors have the same specificity, so if the mouse hovers over the button while it is enabled, the second rule takes precedence. If we want the text to be white in that case, we can reorder the rules like this:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):enabled { color: red }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover { color: white }

Alternatively, we can make the first rule more specific:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover:enabled { color: white }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):enabled { color: red }

A similar issue arises in conjunction with Type Selectors. Consider the following example:

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { color: red }

[QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html) { color: gray }

Both rules apply to [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) instances (since [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) inherits [QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html)) and there is a conflict for the [color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color-prop) property. Because [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) inherits [QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html), it might be tempting to assume that QPushButton is more specific than QAbstractButton. However, for style sheet computations, all Type Selectors have the same specificity, and the rule that appears last takes precedence. In other words, [color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color-prop) is set to gray for all [QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html)s, including [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html)s. If we really want [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html)s to have red text, we can always reorder the rules.

For determining the specificity of a rule, Qt Style Sheets follow the [CSS2 Specification](http://www.w3.org/TR/REC-CSS2/cascade.html" \l "specificity):

*A selector's specificity is calculated as follows:*

* *count the number of ID attributes in the selector (= a)*
* *count the number of other attributes and pseudo-classes in the selector (= b)*
* *count the number of element names in the selector (= c)*
* *ignore pseudo-elements [i.e., [subcontrols](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrols)].*

*Concatenating the three numbers a-b-c (in a number system with a large base) gives the specificity.*

*Some examples:*

\* {} /\* a=0 b=0 c=0 -> specificity = 0 \*/

LI {} /\* a=0 b=0 c=1 -> specificity = 1 \*/

UL LI {} /\* a=0 b=0 c=2 -> specificity = 2 \*/

UL OL+LI {} /\* a=0 b=0 c=3 -> specificity = 3 \*/

H1 + \*[REL=up]{} /\* a=0 b=1 c=1 -> specificity = 11 \*/

UL OL LI.red {} /\* a=0 b=1 c=3 -> specificity = 13 \*/

LI.red.level {} /\* a=0 b=2 c=1 -> specificity = 21 \*/

#x34y {} /\* a=1 b=0 c=0 -> specificity = 100 \*/

## Cascading

Style sheets can be set on the [QApplication](https://doc.qt.io/qt-5/qapplication.html), on parent widgets, and on child widgets. An arbitrary widget's effective style sheet is obtained by merging the style sheets set on the widget's ancestors (parent, grandparent, etc.), as well as any style sheet set on the [QApplication](https://doc.qt.io/qt-5/qapplication.html).

When conflicts arise, the widget's own style sheet is always preferred to any inherited style sheet, irrespective of the specificity of the conflicting rules. Likewise, the parent widget's style sheet is preferred to the grandparent's, etc.

One consequence of this is that setting a style rule on a widget automatically gives it precedence over other rules specified in the ancestor widgets' style sheets or the [QApplication](https://doc.qt.io/qt-5/qapplication.html) style sheet. Consider the following example. First, we set a style sheet on the [QApplication](https://doc.qt.io/qt-5/qapplication.html):

[qApp](https://doc.qt.io/qt-5/qapplication.html" \l "qApp)->setStyleSheet("QPushButton { color: white }");

Then we set a style sheet on a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) object:

myPushButton->setStyleSheet("\* { color: blue }");

The style sheet on the [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) forces the [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) (and any child widget) to have blue text, in spite of the more specific rule set provided by the application-wide style sheet.

The result would have been the same if we had written

myPushButton->setStyleSheet("color: blue");

except that if the [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) had children (which is unlikely), the style sheet would have no impact on them.

Style sheet cascading is a complex topic. Refer to the [CSS2 Specification](http://www.w3.org/TR/CSS2/cascade.html" \l "cascade) for the gory details. Be aware that Qt currently doesn't implement !important.

## Inheritance

In classic CSS, when font and color of an item is not explicitly set, it gets automatically inherited from the parent. By default, when using Qt Style Sheets, a widget does **not** automatically inherit its font and color setting from its parent widget.

For example, consider a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) inside a [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html):

[qApp](https://doc.qt.io/qt-5/qapplication.html" \l "qApp)->setStyleSheet("QGroupBox { color: red; } ");

The [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) does not have an explicit color set. Hence, instead of inheriting color of its parent [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), it has the system color. If we want to set the color on a [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html) and its children, we can write:

[qApp](https://doc.qt.io/qt-5/qapplication.html" \l "qApp)->setStyleSheet("QGroupBox, QGroupBox \* { color: red; }");

In contrast, setting a font and palette using [QWidget::setFont](https://doc.qt.io/qt-5/qwidget.html" \l "font-prop)() and [QWidget::setPalette](https://doc.qt.io/qt-5/qwidget.html" \l "palette-prop)() propagates to child widgets.

If you would prefer that the font and palette propagate to child widgets, you can set the [Qt::AA\_UseStyleSheetPropagationInWidgetStyles](https://doc.qt.io/qt-5/qt.html" \l "ApplicationAttribute-enum) flag, like this:

Usage:

[QCoreApplication](https://doc.qt.io/qt-5/qcoreapplication.html)::setAttribute([Qt](https://doc.qt.io/qt-5/qt.html)::AA\_UseStyleSheetPropagationInWidgetStyles, true);

When the widget-style font and palette propagation is enabled, font and palette changes made through Qt Style Sheets will behave as though the user had manually called the corresponding [QWidget::setPalette](https://doc.qt.io/qt-5/qwidget.html" \l "palette-prop)() and [QWidget::setFont](https://doc.qt.io/qt-5/qwidget.html" \l "font-prop)() methods on all of the QWidgets targeted by the style sheet. If this would have caused propagation in C++, it will cause propagation in style sheets and vice versa.

## Widgets Inside C++ Namespaces

The Type Selector can be used to style widgets of a particular type. For example,

class MyPushButton : public [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) {

// ...

}

// ...

[qApp](https://doc.qt.io/qt-5/qapplication.html" \l "qApp)->setStyleSheet("MyPushButton { background: yellow; }");

Qt Style Sheet uses QObject::className() of the widget to determine when to apply the Type Selector. When custom widgets are inside namespaces, the QObject::className() returns <namespace>[::](https://doc.qt.io/qt-5/qromancalendar.html)<classname>. This conflicts with the syntax for [Sub-Controls](https://doc.qt.io/qt-5/stylesheet-syntax.html" \l "sub-controls). To overcome this problem, when using the Type Selector for widgets inside namespaces, we must replace the "[::](https://doc.qt.io/qt-5/qromancalendar.html)" with "--". For example,

namespace ns {

class MyPushButton : public [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) {

// ...

}

}

// ...

[qApp](https://doc.qt.io/qt-5/qapplication.html" \l "qApp)->setStyleSheet("ns--MyPushButton { background: yellow; }");

## Setting QObject Properties

From 4.3 and above, any designable [Q\_PROPERTY](https://doc.qt.io/qt-5/qobject.html" \l "Q_PROPERTY) can be set using the qproperty-<property name> syntax.

For example,

MyLabel { qproperty-pixmap: url(pixmap.png); }

MyGroupBox { qproperty-titleColor: rgb(100, 200, 100); }

[QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { qproperty-iconSize: 20px 20px; }

If the property references an enum declared with Q\_ENUMS, you should reference its constants by name, i.e., not their numeric value.

**Note:**Use the qproperty syntax with care, as it modifies the widget that is being painted. Also, the qproperty syntax is evaluated only once, which is when the widget is polished by the style. This means that any attempt to use them in pseudo-states such as [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html):hover, will not work.

[Qt Style Sheets](https://doc.qt.io/qt-5/stylesheet.html)[Qt Designer Integration](https://doc.qt.io/qt-5/stylesheet-designer.html)

# Qt Style Sheets Reference

Qt Style Sheets support various properties, pseudo-states, and subcontrols that make it possible to customize the look of widgets.

## List of Stylable Widgets

The following table lists the Qt widgets that can be customized using style sheets:

| Widget | How to Style |
| --- | --- |
| [QAbstractScrollArea](https://doc.qt.io/qt-5/qabstractscrollarea.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  All derivatives of [QAbstractScrollArea](https://doc.qt.io/qt-5/qabstractscrollarea.html), including [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) (all item view classes), support scrollable backgrounds using [background-attachment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-attachment-prop). Setting the background-attachment to fixed provides a background-image that does not scroll with the viewport. Setting the background-attachment to scroll, scrolls the background-image when the scroll bars move.  See [Customizing QAbstractScrollArea](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qabstractscrollarea) for an example. |
| [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The check indicator can be styled using the [::indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "indicator-sub) subcontrol. By default, the indicator is placed in the Top Left corner of the Contents rectangle of the widget.  The [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop) property specifies the spacing between the check indicator and the text.  See [Customizing QCheckBox](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qcheckbox) for an example. |
| [QColumnView](https://doc.qt.io/qt-5/qcolumnview.html) | The grip can be styled be using the [image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "image-prop) property. The arrow indicators can by styled using the [::left-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-arrow-sub) subcontrol and the [::right-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-arrow-sub) subcontrol. |
| [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) | The frame around the combobox can be styled using the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The drop-down button can be styled using the [::drop-down](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "drop-down-sub) subcontrol. By default, the drop-down button is placed in the top right corner of the padding rectangle of the widget. The arrow mark inside the drop-down button can be styled using the [::down-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-arrow-sub) subcontrol. By default, the arrow is placed in the center of the contents rectangle of the drop-down subcontrol.  See [Customizing QComboBox](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qcombobox) for an example. |
| [QDateEdit](https://doc.qt.io/qt-5/qdateedit.html) | See [QSpinBox](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qspinbox-widget). |
| [QDateTimeEdit](https://doc.qt.io/qt-5/qdatetimeedit.html) | See [QSpinBox](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qspinbox-widget). |
| [QDialog](https://doc.qt.io/qt-5/qdialog.html) | Supports only the [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop), [background-clip](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-clip-prop) and [background-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-origin-prop) properties.  **Warning:** Make sure you define the [Q\_OBJECT](https://doc.qt.io/qt-5/qobject.html" \l "Q_OBJECT) macro for your custom widget. |
| [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) | The layout of buttons can be altered using the [button-layout](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "button-layout-prop) property. |
| [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) | Supports styling of the title bar and the title bar buttons when docked.  The dock widget border can be styled using the [border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-prop) property. The [::title](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "title-sub) subcontrol can be used to customize the title bar. The close and float buttons are positioned with respect to the [::title](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "title-sub) subcontrol using the [::close-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "close-button-sub) and [::float-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "float-button-sub) respectively.  When the title bar is vertical, the [:vertical](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "vertical-ps) pseudo class is set. In addition, depending on [QDockWidget::DockWidgetFeature](https://doc.qt.io/qt-5/qdockwidget.html" \l "DockWidgetFeature-enum), the [:closable](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "closable-ps), [:floatable](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "floatable-ps) and [:movable](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "movable-ps) pseudo states are set.  **Note:**Use QMainWindow::separator to style the resize handle.  **Warning:** The style sheet has no effect when the [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) is undocked as Qt uses native top level windows when undocked.  See [Customizing QDockWidget](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qdockwidget) for an example. |
| [QDoubleSpinBox](https://doc.qt.io/qt-5/qdoublespinbox.html) | See [QSpinBox](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qspinbox-widget). |
| [QFrame](https://doc.qt.io/qt-5/qframe.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  Since 4.3, setting a stylesheet on a [QLabel](https://doc.qt.io/qt-5/qlabel.html) automatically sets the [QFrame::frameStyle](https://doc.qt.io/qt-5/qframe.html" \l "frameStyle) property to [QFrame::StyledPanel](https://doc.qt.io/qt-5/qframe.html" \l "Shape-enum).  See [Customizing QFrame](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qframe) for an example. |
| [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The title can be styled using the [::title](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "title-sub) subcontrol. By default, the title is placed depending on QGroupBox::textAlignment.  In the case of a checkable [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), the title includes the check indicator. The indicator is styled using the [::indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "indicator-sub) subcontrol. The [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop) property can be used to control the spacing between the text and indicator.  See [Customizing QGroupBox](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qgroupbox) for an example. |
| [QHeaderView](https://doc.qt.io/qt-5/qheaderview.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The sections of the header view are styled using the [::section](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "section-sub) sub control. The section Sub-control supports the [:middle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "middle-ps), [:first](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "first-ps), [:last](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "last-ps), [:only-one](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "only-one-ps), [:next-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "next-selected-ps), [:previous-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "previous-selected-ps), [:selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selected-ps), and [:checked](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "checked-ps) pseudo states.  The sort indicator can be styled using the [::up-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "up-arrow-sub) and the [::down-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-arrow-sub) Sub-control.  See [Customizing QHeaderView](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qheaderview) for an example. |
| [QLabel](https://doc.qt.io/qt-5/qlabel.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). Does not support the [:hover](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "hover-ps) pseudo-state.  Since 4.3, setting a stylesheet on a [QLabel](https://doc.qt.io/qt-5/qlabel.html) automatically sets the [QFrame::frameStyle](https://doc.qt.io/qt-5/qframe.html" \l "frameStyle) property to [QFrame::StyledPanel](https://doc.qt.io/qt-5/qframe.html" \l "Shape-enum).  See [Customizing QFrame](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qframe) for an example (a [QLabel](https://doc.qt.io/qt-5/qlabel.html) derives from [QFrame](https://doc.qt.io/qt-5/qframe.html)). |
| [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) | Support the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  The color and background of the selected item is styled using [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) respectively.  The password character can be styled using the [lineedit-password-character](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "lineedit-password-character-prop) property.  The password mask delay can be changed using the [lineedit-password-mask-delay](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "lineedit-password-mask-delay-prop)  See [Customizing QLineEdit](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qlineedit) for an example. |
| [QListView](https://doc.qt.io/qt-5/qlistview.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). When [alternating row colors](https://doc.qt.io/qt-5/qabstractitemview.html" \l "alternatingRowColors-prop) is enabled, the alternating colors can be styled using the [alternate-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alternate-background-color-prop) property.  The color and background of the selected item is styled using [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) respectively.  The selection behavior is controlled by the [show-decoration-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "show-decoration-selected-prop) property.  Use the [::item](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "item-sub) subcontrol for more fine grained control over the items in the [QListView](https://doc.qt.io/qt-5/qlistview.html).  See [QAbsractScrollArea](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qabstractscrollarea-widget) to style scrollable backgrounds.  See [Customzing QListView](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qlistview) for an example. |
| [QListWidget](https://doc.qt.io/qt-5/qlistwidget.html) | See [QListView](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qlistview-widget). |
| [QMainWindow](https://doc.qt.io/qt-5/qmainwindow.html) | Supports styling of the separator  The separator in a [QMainWindow](https://doc.qt.io/qt-5/qmainwindow.html) when using [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) is styled using the [::separator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "separator-sub) subcontrol.  See [Customizing QMainWindow](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qmainwindow) for an example. |
| [QMenu](https://doc.qt.io/qt-5/qmenu.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  Individual items are styled using the [::item](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "item-sub) subcontrol. In addition to the usually supported pseudo states, item subcontrol supports the [:selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selected-ps), [:default](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "default-ps), [:exclusive](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "exclusive-ps) and the [non-exclusive](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "non-exclusive-ps) pseudo states.  The indicator of checkable menu items is styled using the [::indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "indicator-sub) subcontrol.  The separator is styled using the [::separator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "separator-sub) subcontrol.  For items with a sub menu, the arrow marks are styled using the [right-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-arrow-sub) and [left-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-arrow-sub).  The scroller is styled using the [::scroller](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "scroller-sub).  The tear-off is styled using the [::tearoff](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tearoff-sub).  See [Customizing QMenu](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qmenu) for an example. |
| [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop) property specifies the spacing between menu items. Individual items are styled using the [::item](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "item-sub) subcontrol.  **Warning:** When running on Qt/Mac, the menu bar is usually embedded into the system-wide menu bar. In this case, the style sheet will have no effect.  See [Customizing QMenuBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qmenubar) for an example. |
| [QMessageBox](https://doc.qt.io/qt-5/qmessagebox.html) | The [messagebox-text-interaction-flags](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "messagebox-text-interaction-flags-prop) property can be used to alter the interaction with text in the message box. |
| [QProgressBar](https://doc.qt.io/qt-5/qprogressbar.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The chunks of the progress bar can be styled using the [::chunk](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "chunk-sub) subcontrol. The chunk is displayed on the Contents rectangle of the widget.  If the progress bar displays text, use the [text-align](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "text-align-prop) property to position the text.  Indeterminate progress bars have the [:indeterminate](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "indeterminate-ps) pseudo state set.  See [Customizing QProgressBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qprogressbar) for an example. |
| [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). Supports the [:default](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "default-ps), [:flat](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "flat-ps), [:checked](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "checked-ps) pseudo states.  For [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) with a menu, the menu indicator is styled using the [::menu-indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "menu-indicator-sub) subcontrol. Appearance of checkable push buttons can be customized using the [:open](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "open-ps) and [:closed](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "closed-ps) pseudo-states.  **Warning:** If you only set a background-color on a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), the background may not appear unless you set the border property to some value. This is because, by default, the [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) draws a native border which completely overlaps the background-color. For example,  [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { background-color: red; border: none; }  See [Customizing QPushButton](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qpushbutton) for an example. |
| [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The check indicator can be styled using the [::indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "indicator-sub) subcontrol. By default, the indicator is placed in the Top Left corner of the Contents rectangle of the widget.  The [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop) property specifies the spacing between the check indicator and the text.  See [Customizing QRadioButton](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qradiobutton) for an example. |
| [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The Contents rectangle of the widget is considered to be the groove over which the slider moves. The extent of the [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html) (i.e the width or the height depending on the orientation) is set using the [width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "width-prop) or [height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "height-prop) property respectively. To determine the orientation, use the [:horizontal](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "horizontal-ps) and the [:vertical](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "vertical-ps) pseudo states.  The slider can be styled using the [::handle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "handle-sub) subcontrol. Setting the [min-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-width-prop) or [min-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-height-prop) provides size contraints for the slider depending on the orientation.  The [::add-line](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "add-line-sub) subcontrol can be used to style the button to add a line. By default, the add-line subcontrol is placed in top right corner of the Border rectangle of the widget. Depending on the orientation the [::right-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-arrow-sub) or [::down-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-arrow-sub). By default, the arrows are placed in the center of the Contents rectangle of the add-line subcontrol.  The [::sub-line](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "sub-line-sub) subcontrol can be used to style the button to subtract a line. By default, the sub-line subcontrol is placed in bottom right corner of the Border rectangle of the widget. Depending on the orientation the [::left-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-arrow-sub) or [::up-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "up-arrow-sub). By default, the arrows are placed in the center of the Contents rectangle of the sub-line subcontrol.  The [::sub-page](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "sub-page-sub) subcontrol can be used to style the region of the slider that subtracts a page. The [::add-page](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "add-page-sub) subcontrol can be used to style the region of the slider that adds a page.  See [Customizing QScrollBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qscrollbar) for an example. |
| [QSizeGrip](https://doc.qt.io/qt-5/qsizegrip.html) | Supports the [width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "width-prop), [height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "height-prop), and [image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "image-prop) properties.  See [Customizing QSizeGrip](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qsizegrip) for an example. |
| [QSlider](https://doc.qt.io/qt-5/qslider.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). For horizontal slides, the [min-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-width-prop) and [height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "height-prop) properties must be provided. For vertical sliders, the [min-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-height-prop) and [width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "width-prop) properties must be provided.  The groove of the slider is styled using the [::groove](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "groove-sub). The groove is positioned by default in the Contents rectangle of the widget. The thumb of the slider is styled using [::handle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "handle-sub) subcontrol. The subcontrol moves in the Contents rectangle of the groove subcontrol.  See [Customizing QSlider](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qslider) for an example. |
| [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html) | The frame of the spin box can be styled using the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  The up button and arrow can be styled using the [::up-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "up-button-sub) and [::up-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "up-arrow-sub) subcontrols. By default, the up-button is placed in the top right corner in the Padding rectangle of the widget. Without an explicit size, it occupies half the height of its reference rectangle. The up-arrow is placed in the center of the Contents rectangle of the up-button.  The down button and arrow can be styled using the [::down-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-button-sub) and [::down-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-arrow-sub) subcontrols. By default, the down-button is placed in the bottom right corner in the Padding rectangle of the widget. Without an explicit size, it occupies half the height of its reference rectangle. The bottom-arrow is placed in the center of the Contents rectangle of the bottom-button.  See [Customizing QSpinBox](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qspinbox) for an example. |
| [QSplitter](https://doc.qt.io/qt-5/qsplitter.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The handle of the splitter is styled using the [::handle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "handle-sub) subcontrol.  See [Customizing QSplitter](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qsplitter) for an example. |
| [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html) | Supports only the [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop) property. The frame for individual items can be style using the [::item](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "item-sub) subcontrol.  See [Customizing QStatusBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qstatusbar) for an example. |
| [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) | Individual tabs may be styled using the [::tab](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tab-sub) subcontrol. Close buttons using the [::close-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "close-button-sub) The tabs support the [:only-one](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "only-one-ps), [:first](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "first-ps), [:last](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "last-ps), [:middle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "middle-ps), [:previous--selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "previous-selected-ps), [:next-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "next-selected-ps), [:selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selected-ps) pseudo states.  The [:top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-ps), [:left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-ps), [:right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-ps), [:bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-ps) pseudo states depending on the orientation of the tabs.  Overlapping tabs for the selected state are created by using negative margins or using the absolute position scheme.  The tear indicator of the [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) is styled using the [::tear](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tear-sub) subcontrol.  [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) used two QToolButtons for its scrollers that can be styled using the QTabBar QToolButton selector. To specify the width of the scroll button use the [::scroller](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "scroller-sub) subcontrol.  The alignment of the tabs within the [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) is styled using the [alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) property.  **Warning:**  To change the position of the [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) within a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html), use the [tab-bar](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tab-bar-sub) subcontrol (and set subcontrol-position).  See [Customizing QTabBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtabwidget-and-qtabbar) for an example. |
| [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html) | The frame of the tab widget is styled using the [::pane](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "pane-sub) subcontrol. The left and right corners are styled using the [::left-corner](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-corner-sub) and [::right-corner](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-corner-sub) respectively. The position of the tab bar is controlled using the [::tab-bar](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tab-bar-sub) subcontrol.  By default, the subcontrols have positions of a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html) in the QWindowsStyle. To place the [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) in the center, set the subcontrol-position of the tab-bar subcontrol.  The [:top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-ps), [:left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-ps), [:right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-ps), [:bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-ps) pseudo states depending on the orientation of the tabs.  See [Customizing QTabWidget](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtabwidget-and-qtabbar) for an example. |
| [QTableView](https://doc.qt.io/qt-5/qtableview.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). When [alternating row colors](https://doc.qt.io/qt-5/qabstractitemview.html" \l "alternatingRowColors-prop) is enabled, the alternating colors can be styled using the [alternate-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alternate-background-color-prop) property.  The color and background of the selected item is styled using [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) respectively.  The corner widget in a [QTableView](https://doc.qt.io/qt-5/qtableview.html) is implemented as a [QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html) and can be styled using the "[QTableView](https://doc.qt.io/qt-5/qtableview.html) QTableCornerButton::section" selector.  **Warning:** If you only set a background-color on a QTableCornerButton, the background may not appear unless you set the border property to some value. This is because, by default, the QTableCornerButton draws a native border which completely overlaps the background-color.  The color of the grid can be specified using the [gridline-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "gridline-color-prop) property.  See [QAbsractScrollArea](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qabstractscrollarea-widget) to style scrollable backgrounds.  See [Customzing QTableView](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtableview) for an example. |
| [QTableWidget](https://doc.qt.io/qt-5/qtablewidget.html) | See [QTableView](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qtableview-widget). |
| [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  The color and background of selected text is styled using [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) respectively.  See [QAbsractScrollArea](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qabstractscrollarea-widget) to style scrollable backgrounds. |
| [QTimeEdit](https://doc.qt.io/qt-5/qtimeedit.html) | See [QSpinBox](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qspinbox-widget). |
| [QToolBar](https://doc.qt.io/qt-5/qtoolbar.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  The [:top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-ps), [:left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-ps), [:right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-ps), [:bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-ps) pseudo states depending on the area in which the tool bar is grouped.  The [:first](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "first-ps), [:last](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "last-ps), [:middle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "middle-ps), [:only-one](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "only-one-ps) pseudo states indicator the position of the tool bar within a line group (See [QStyleOptionToolBar::positionWithinLine](https://doc.qt.io/qt-5/qstyleoptiontoolbar.html" \l "positionWithinLine-var)).  The separator of a [QToolBar](https://doc.qt.io/qt-5/qtoolbar.html) is styled using the [::separator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "separator-sub) subcontrol.  The handle (to move the toolbar) is styled using the [::handle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "handle-sub) subcontrol.  See [Customizing QToolBar](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtoolbar) for an example. |
| [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  If the [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) has a menu, is [::menu-indicator](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "menu-indicator-sub) subcontrol can be used to style the indicator. By default, the menu-indicator is positioned at the bottom right of the Padding rectangle of the widget.  If the [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) is in [QToolButton::MenuButtonPopup](https://doc.qt.io/qt-5/qtoolbutton.html" \l "ToolButtonPopupMode-enum) mode, the [::menu-button](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "menu-button-sub) subcontrol is used to draw the menu button. [::menu-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "menu-arrow-sub) subcontrol is used to draw the menu arrow inside the menu-button. By default, it is positioned in the center of the Contents rectangle of the menu-button subcontrol.  When the [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) displays arrows, the [::up-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "up-arrow-sub), [::down-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "down-arrow-sub), [::left-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-arrow-sub) and [::right-arrow](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-arrow-sub) subcontrols are used.  **Warning:** If you only set a background-color on a [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html), the background will not appear unless you set the border property to some value. This is because, by default, the [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) draws a native border which completely overlaps the background-color. For example,  [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) { background-color: red; border: none; }  See [Customizing QToolButton](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtoolbutton) for an example. |
| [QToolBox](https://doc.qt.io/qt-5/qtoolbox.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model).  The individual tabs can by styled using the [::tab](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tab-sub) subcontrol. The tabs support the [:only-one](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "only-one-ps), [:first](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "first-ps), [:last](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "last-ps), [:middle](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "middle-ps), [:previous-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "previous-selected-ps), [:next-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "next-selected-ps), [:selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selected-ps) pseudo states. |
| [QToolTip](https://doc.qt.io/qt-5/qtooltip.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). The [opacity](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "opacity-prop) property controls the opacity of the tooltip.  See [Customizing QFrame](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qframe) for an example (a [QToolTip](https://doc.qt.io/qt-5/qtooltip.html) is a [QFrame](https://doc.qt.io/qt-5/qframe.html)). |
| [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) | Supports the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "box-model). When [alternating row colors](https://doc.qt.io/qt-5/qabstractitemview.html" \l "alternatingRowColors-prop) is enabled, the alternating colors can be styled using the [alternate-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alternate-background-color-prop) property.  The color and background of the selected item is styled using [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) respectively.  The selection behavior is controlled by the [show-decoration-selected](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "show-decoration-selected-prop) property.  The branches of the tree view can be styled using the [::branch](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "branch-sub) subcontrol. The ::branch Sub-control supports the [:open](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "open-ps), [:closed](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "closed-ps), [:has-sibling](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "has-siblings-ps) and [:has-children](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "has-children-ps) pseudo states.  Use the [::item](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "item-sub) subcontrol for more fine grained control over the items in the [QTreeView](https://doc.qt.io/qt-5/qtreeview.html).  See [QAbsractScrollArea](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qabstractscrollarea-widget) to style scrollable backgrounds.  See [Customizing QTreeView](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-qtreeview) for an example to style the branches. |
| [QTreeWidget](https://doc.qt.io/qt-5/qtreewidget.html) | See [QTreeView](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "qtreeview-widget). |
| [QWidget](https://doc.qt.io/qt-5/qwidget.html) | Supports only the [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop), [background-clip](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-clip-prop) and [background-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-origin-prop) properties.  If you subclass from [QWidget](https://doc.qt.io/qt-5/qwidget.html), you need to provide a paintEvent for your custom [QWidget](https://doc.qt.io/qt-5/qwidget.html) as below:  void CustomWidget::paintEvent([QPaintEvent](https://doc.qt.io/qt-5/qpaintevent.html) \*)  {  [QStyleOption](https://doc.qt.io/qt-5/qstyleoption.html) opt;  opt.init(this);  [QPainter](https://doc.qt.io/qt-5/qpainter.html) p(this);  style()->drawPrimitive([QStyle](https://doc.qt.io/qt-5/qstyle.html)::PE\_Widget, &opt, &p, this);  }  The above code is a no-operation if there is no stylesheet set.  **Warning:** Make sure you define the [Q\_OBJECT](https://doc.qt.io/qt-5/qobject.html" \l "Q_OBJECT) macro for your custom widget. |

## List of Properties

The table below lists all the properties supported by Qt Style Sheets. Which values can be given to an property depend on the [property's type](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "list-of-property-types). Unless otherwise specified, properties below apply to all widgets. Properties marked with an asterisk \* are specific to Qt and have no equivalent in CSS2 or CSS3.

| Property | Type | Description |
| --- | --- | --- |
| **alternate-background-color** | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The [alternate background color](https://doc.qt.io/qt-5/qabstractitemview.html" \l "alternatingRowColors-prop) used in [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses.  If this property is not set, the default value is whatever is set for the palette's [AlternateBase](https://doc.qt.io/qt-5/qpalette.html" \l "ColorRole-enum) role.  Example:  [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) {  alternate-background-color: blue;  background: yellow;  }  See also [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop) and [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop). |
| **background** | [Background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background) | Shorthand notation for setting the background. Equivalent to specifying background-color, background-image, background-repeat, and/or background-position.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QDialog](https://doc.qt.io/qt-5/qdialog.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), [QToolTip](https://doc.qt.io/qt-5/qtooltip.html), and plain [QWidget](https://doc.qt.io/qt-5/qwidget.html)s.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { background: yellow }  Often, it is required to set a fill pattern similar to the styles in [Qt::BrushStyle](https://doc.qt.io/qt-5/qt.html" \l "BrushStyle-enum). You can use the background-color property for [Qt::SolidPattern](https://doc.qt.io/qt-5/qt.html" \l "BrushStyle-enum), [Qt::RadialGradientPattern](https://doc.qt.io/qt-5/qt.html" \l "BrushStyle-enum), [Qt::LinearGradientPattern](https://doc.qt.io/qt-5/qt.html" \l "BrushStyle-enum) and [Qt::ConicalGradientPattern](https://doc.qt.io/qt-5/qt.html" \l "BrushStyle-enum). The other patterns are easily achieved by creating a background image that contains the pattern.  Example:  [QLabel](https://doc.qt.io/qt-5/qlabel.html) {  background-image: url(dense6pattern.png);  background-repeat: repeat-xy;  }  See also [background-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-origin-prop), [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop), [background-clip](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-clip-prop), [background-attachment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-attachment-prop) and [alternate-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alternate-background-color-prop). |
| background-color | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The background color used for the widget.  Examples:  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { background-color: yellow }  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) { background-color: rgb(255, 0, 0) } |
| background-image | [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url) | The background image used for the widget. Semi-transparent parts of the image let the background-color shine through.  Example:  [QFrame](https://doc.qt.io/qt-5/qframe.html) { background-image: url(:/images/hydro.png) } |
| background-repeat | [Repeat](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "repeat) | Whether and how the background image is repeated to fill the background-origin rectangle.  If this property is not specified, the background image is repeated in both directions (repeat).  Example:  [QFrame](https://doc.qt.io/qt-5/qframe.html) {  background: white url(:/images/ring.png);  background-repeat: repeat-y;  background-position: left;  } |
| background-position | [Alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) | The alignment of the background image within the background-origin rectangle.  If this property is not specified, the alignment is top left.  Example:  [QFrame](https://doc.qt.io/qt-5/qframe.html) {  background: url(:/images/footer.png);  background-position: bottom left;  } |
| **background-attachment** | [Attachment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "attachment) | Determines whether the background-image in a [QAbstractScrollArea](https://doc.qt.io/qt-5/qabstractscrollarea.html) is scrolled or fixed with respect to the viewport. By default, the background-image scrolls with the viewport.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) {  background-image: url("leaves.png");  background-attachment: fixed;  }  See also [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop) |
| **background-clip** | [Origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "origin) | The widget's rectangle, in which the background is drawn.  This property specifies the rectangle to which the background-color and background-image are clipped.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QDialog](https://doc.qt.io/qt-5/qdialog.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), [QToolTip](https://doc.qt.io/qt-5/qtooltip.html), and plain [QWidget](https://doc.qt.io/qt-5/qwidget.html)s.  If this property is not specified, the default is border.  Example:  [QFrame](https://doc.qt.io/qt-5/qframe.html) {  background-image: url(:/images/header.png);  background-position: top left;  background-origin: content;  background-clip: padding;  }  See also [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop), [background-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-origin-prop) and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| **background-origin** | [Origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "origin) | The widget's background rectangle, to use in conjunction with background-position and background-image.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QDialog](https://doc.qt.io/qt-5/qdialog.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), [QToolTip](https://doc.qt.io/qt-5/qtooltip.html), and plain [QWidget](https://doc.qt.io/qt-5/qwidget.html)s.  If this property is not specified, the default is padding.  Example:  [QFrame](https://doc.qt.io/qt-5/qframe.html) {  background-image: url(:/images/header.png);  background-position: top left;  background-origin: content;  }  See also [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop) and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| **border** | [Border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border) | Shorthand notation for setting the widget's border. Equivalent to specifying border-color, border-style, and/or border-width.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), [QToolTip](https://doc.qt.io/qt-5/qtooltip.html), and plain [QWidget](https://doc.qt.io/qt-5/qwidget.html)s.  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) { border: 1px solid white } |
| border-top | [Border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border) | Shorthand notation for setting the widget's top border. Equivalent to specifying border-top-color, border-top-style, and/or border-top-width. |
| border-right | [Border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border) | Shorthand notation for setting the widget's right border. Equivalent to specifying border-right-color, border-right-style, and/or border-right-width. |
| border-bottom | [Border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border) | Shorthand notation for setting the widget's bottom border. Equivalent to specifying border-bottom-color, border-bottom-style, and/or border-bottom-width. |
| border-left | [Border](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border) | Shorthand notation for setting the widget's left border. Equivalent to specifying border-left-color, border-left-style, and/or border-left-width. |
| **border-color** | [Box Colors](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "box-colors) | The color of all the border's edges. Equivalent to specifying border-top-color, border-right-color, border-bottom-color, and border-left-color.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), [QToolTip](https://doc.qt.io/qt-5/qtooltip.html), and plain [QWidget](https://doc.qt.io/qt-5/qwidget.html)s.  If this property is not specified, it defaults to [color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color-prop) (i.e., the widget's foreground color).  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) {  border-width: 1px;  border-style: solid;  border-color: white;  }  See also [border-style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style-prop), [border-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-width-prop), [border-image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-image-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| border-top-color | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The color of the border's top edge. |
| border-right-color | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The color of the border's right edge. |
| border-bottom-color | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The color of the border's bottom edge. |
| border-left-color | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The color of the border's left edge. |
| **border-image** | [Border Image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-image) | The image used to fill the border. The image is cut into nine parts and stretched appropriately if necessary. See [Border Image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-image) for details.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  See also [border-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-color-prop), [border-style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style-prop), [border-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-width-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| **border-radius** | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius of the border's corners. Equivalent to specifying border-top-left-radius, border-top-right-radius, border-bottom-right-radius, and border-bottom-left-radius.  The border-radius clips the element's [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop).  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, it defaults to 0.  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) {  border-width: 1px;  border-style: solid;  border-radius: 4px;  }  See also [border-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-width-prop) and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| border-top-left-radius | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius of the border's top-left corner. |
| border-top-right-radius | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius of the border's top-right corner. |
| border-bottom-right-radius | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius of the border's bottom-right corner. Setting this property to a positive value results in a rounded corner. |
| border-bottom-left-radius | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius of the border's bottom-left corner. Setting this property to a positive value results in a rounded corner. |
| **border-style** | [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | The style of all the border's edges.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, it defaults to none.  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) {  border-width: 1px;  border-style: solid;  border-color: blue;  }  See also [border-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-color-prop), [border-style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style-prop), [border-image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-image-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| border-top-style | [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | The style of the border's top edge. |
| border-right-style | [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | The style of the border's right edge/ |
| border-bottom-style | [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | The style of the border's bottom edge. |
| border-left-style | [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | The style of the border's left edge. |
| **border-width** | [Box Lengths](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "box-lengths) | The width of the border. Equivalent to setting border-top-width, border-right-width, border-bottom-width, and border-left-width.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) {  border-width: 2px;  border-style: solid;  border-color: darkblue;  }  See also [border-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-color-prop), [border-radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-radius-prop), [border-style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style-prop), [border-image](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-image-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| border-top-width | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width of the border's top edge. |
| border-right-width | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width of the border's right edge. |
| border-bottom-width | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width of the border's bottom edge. |
| border-left-width | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width of the border's left edge. |
| **bottom** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is relative (the default), moves a subcontrol by a certain offset up; specifying bottom: *y* is then equivalent to specifying [top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-prop): -*y*.  If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is absolute, the bottom property specifies the subcontrol's bottom edge in relation to the parent's bottom edge (see also [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop)).  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button { bottom: 2px }  See also [left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-prop), [right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-prop), and [top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-prop). |
| **button-layout** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | The layout of buttons in a [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) or a [QMessageBox](https://doc.qt.io/qt-5/qmessagebox.html). The possible values are 0 ([WinLayout](https://doc.qt.io/qt-5/qdialogbuttonbox.html" \l "ButtonLayout-enum)), 1 ([MacLayout](https://doc.qt.io/qt-5/qdialogbuttonbox.html" \l "ButtonLayout-enum)), 2 ([KdeLayout](https://doc.qt.io/qt-5/qdialogbuttonbox.html" \l "ButtonLayout-enum)), 3 ([GnomeLayout](https://doc.qt.io/qt-5/qdialogbuttonbox.html" \l "ButtonLayout-enum)) and 5 ([AndroidLayout](https://doc.qt.io/qt-5/qdialogbuttonbox.html" \l "ButtonLayout-enum)).  If this property is not specified, it defaults to the value specified by the current style for the [SH\_DialogButtonLayout](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  \* { button-layout: 2 } |
| **color** | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The color used to render text.  This property is supported by all widgets that respect the [QWidget::palette](https://doc.qt.io/qt-5/qwidget.html" \l "palette-prop).  If this property is not set, the default is whatever is set for in the widget's palette for the [QWidget::foregroundRole](https://doc.qt.io/qt-5/qwidget.html" \l "foregroundRole) (typically black).  Example:  [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { color: red }  See also [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop) and [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop). |
| **dialogbuttonbox-buttons-have-icons** | [Boolean](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "boolean) | Whether the buttons in a [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) show icons  If this property is set to 1, the buttons of a [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) show icons; if it is set to 0, the icons are not shown.  See the [List of Icons](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "list-of-icons) section for information on how to set icons.  [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) { dialogbuttonbox-buttons-have-icons: 1; }  **Note:**Styles defining this property must be applied before the [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html) is created; this means that you must apply the style to the parent widget or to the application itself. |
| **font** | [Font](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font) | Shorthand notation for setting the text's font. Equivalent to specifying font-family, font-size, font-style, and/or font-weight.  This property is supported by all widgets that respect the [QWidget::font](https://doc.qt.io/qt-5/qwidget.html" \l "font-prop).  If this property is not set, the default is the [QWidget::font](https://doc.qt.io/qt-5/qwidget.html" \l "font-prop).  Example:  [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html) { font: bold italic large "Times New Roman" } |
| font-family | String | The font family.  Example:  [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html) { font-family: "New Century Schoolbook" } |
| font-size | [Font Size](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-size) | The font size. In this version of Qt, only pt and px metrics are supported.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { font-size: 12px } |
| font-style | [Font Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-style) | The font style.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { font-style: italic } |
| font-weight | [Font Weight](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-weight) | The weight of the font. |
| **gridline-color**\* | [Color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color) | The color of the grid line in a [QTableView](https://doc.qt.io/qt-5/qtableview.html).  If this property is not specified, it defaults to the value specified by the current style for the [SH\_Table\_GridLineColor](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  \* { gridline-color: gray } |
| **height** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The height of a subcontrol (or in some case, a widget).  If this property is not specified, it defaults to a value that depends on the subcontrol/widget and on the current style.  **Warning:** Unless otherwise specified, this property has no effect when set on widgets. If you want a widget with a fixed height, set the [min-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-width-prop) and [max-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "max-width-prop) to the same value.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button { height: 10px }  See also [width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "width-prop). |
| **icon-size** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width and height of the icon in a widget.  The icon size of the following widgets can be set using this property.   * [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html) * [QListView](https://doc.qt.io/qt-5/qlistview.html) * [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) * [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html) * [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) * [QToolBar](https://doc.qt.io/qt-5/qtoolbar.html) * [QToolBox](https://doc.qt.io/qt-5/qtoolbox.html) * [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) |
| **image**\* | [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url)+ | The image that is drawn in the contents rectangle of a subcontrol.  The image property accepts a list of [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url)s or an svg. The actual image that is drawn is determined using the same algorithm as [QIcon](https://doc.qt.io/qt-5/qicon.html) (i.e) the image is never scaled up but always scaled down if necessary. If a svg is specified, the image is scaled to the size of the contents rectangle.  Setting the image property on sub controls implicitly sets the width and height of the sub-control (unless the image in a SVG).  In Qt 4.3 and later, the alignment of the image within the rectangle can be specified using [image-position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "image-position-prop).  This property is for subcontrols only--we don't support it for other elements.  **Warning:** The [QIcon](https://doc.qt.io/qt-5/qicon.html) SVG plugin is needed to render SVG images.  Example:  // implicitly sets the size of down-button to the  // size of spindown.png  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button { image: url(:/images/spindown.png) } |
| **image-position** | [alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) | In Qt 4.3 and later, the alignment of the image image's position can be specified using relative or absolute position. |
| **left** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is relative (the default), moves a subcontrol by a certain offset to the right.  If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is absolute, the left property specifies the subcontrol's left edge in relation to the parent's left edge (see also [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop)).  If this property is not specified, it defaults to 0.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button { left: 2px }  See also [right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-prop), [top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-prop), and [bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-prop). |
| **lineedit-password-character\*** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | The [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) password character as a Unicode number.  If this property is not specified, it defaults to the value specified by the current style for the [SH\_LineEdit\_PasswordCharacter](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  \* { lineedit-password-character: 9679 } |
| **lineedit-password-mask-delay\*** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | The [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) password mask delay in milliseconds before [lineedit-password-character](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "lineedit-password-character-prop) is applied to visible character.  If this property is not specified, it defaults to the value specified by the current style for the [SH\_LineEdit\_PasswordMaskDelay](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  **This property was added in Qt 5.4.**  Example:  \* { lineedit-password-mask-delay: 1000 } |
| **margin** | [Box Lengths](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "box-lengths) | The widget's margins. Equivalent to specifying margin-top, margin-right, margin-bottom, and margin-left.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, it defaults to 0.  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) { margin: 2px }  See also [padding](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "padding-prop), [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| margin-top | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's top margin. |
| margin-right | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's right margin. |
| margin-bottom | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's bottom margin. |
| margin-left | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's left margin. |
| **max-height** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's or a subcontrol's maximum height.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSizeGrip](https://doc.qt.io/qt-5/qsizegrip.html), [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  The value is relative to the contents rect in the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model).  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html) { max-height: 24px }  See also [max-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "max-width-prop). |
| **max-width** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's or a subcontrol's maximum width.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSizeGrip](https://doc.qt.io/qt-5/qsizegrip.html), [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  The value is relative to the contents rect in the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model).  Example:  [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) { max-width: 72px }  See also [max-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "max-height-prop). |
| **messagebox-text-interaction-flags\*** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | The interaction behavior for text in a message box. Possible values are based on [Qt::TextInteractionFlags](https://doc.qt.io/qt-5/qt.html" \l "TextInteractionFlag-enum).  If this property is not specified, it defaults to the value specified by the current style for the [SH\_MessageBox\_TextInteractionFlags](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  [QMessageBox](https://doc.qt.io/qt-5/qmessagebox.html) { messagebox-text-interaction-flags: 5 } |
| **min-height** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's or a subcontrol's minimum height.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSizeGrip](https://doc.qt.io/qt-5/qsizegrip.html), [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, the minimum height is derived based on the widget's contents and the style.  The value is relative to the contents rect in the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model).  Example:  [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) { min-height: 24px }  See also [min-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-width-prop). |
| **min-width** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's or a subcontrol's minimum width.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSizeGrip](https://doc.qt.io/qt-5/qsizegrip.html), [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, the minimum width is derived based on the widget's contents and the style.  The value is relative to the contents rect in the [box model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model).  Example:  [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) { min-width: 72px }  See also [min-height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-height-prop). |
| **opacity\*** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | The opacity for a widget. Possible values are from 0 (transparent) to 255 (opaque). For the moment, this is only supported for [tooltips](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, it defaults to the value specified by the current style for the [SH\_ToolTipLabel\_Opacity](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  [QToolTip](https://doc.qt.io/qt-5/qtooltip.html) { opacity: 223 } |
| **outline** |  | The outline drawn around the object's border. |
| **outline-color** | [Color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color) | The color of the outline. See also [border-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-color-prop) |
| **outline-offset** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The outline's offset from the border of the widget. |
| **outline-style** |  | Specifies the pattern used to draw the outline. See also [border-style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style-prop) |
| **outline-radius** |  | Adds rounded corners to the outline |
| **outline-bottom-left-radius** | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius for the bottom-left rounded corner of the outline. |
| **outline-bottom-right-radius** | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius for the bottom-right rounded corner of the outline. |
| **outline-top-left-radius** | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius for the top-left corner of the outline. |
| **outline-top-right-radius** | [Radius](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "radius) | The radius for the top-right rounded corner of the outline. |
| **padding** | [Box Lengths](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "box-lengths) | The widget's padding. Equivalent to specifying padding-top, padding-right, padding-bottom, and padding-left.  This property is supported by [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) subclasses, [QAbstractSpinBox](https://doc.qt.io/qt-5/qabstractspinbox.html) subclasses, [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QFrame](https://doc.qt.io/qt-5/qframe.html), [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html), [QLabel](https://doc.qt.io/qt-5/qlabel.html), [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html), [QMenu](https://doc.qt.io/qt-5/qmenu.html), [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html), [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html), and [QToolTip](https://doc.qt.io/qt-5/qtooltip.html).  If this property is not specified, it defaults to 0.  Example:  [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) { padding: 3px }  See also [margin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "margin-prop), [spacing](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "spacing-prop), and [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| padding-top | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's top padding. |
| padding-right | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's right padding. |
| padding-bottom | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's bottom padding. |
| padding-left | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The widget's left padding. |
| **paint-alternating-row-colors-for-empty-area** | bool | Whether the [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) paints alternating row colors for the empty area (i.e the area where there are no items) |
| **position** | relative | absolute | Whether offsets specified using [left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-prop), [right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-prop), [top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-prop), and [bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-prop) are relative or absolute coordinates.  If this property is not specified, it defaults to relative. |
| **right** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is relative (the default), moves a subcontrol by a certain offset to the left; specifying right: *x* is then equivalent to specifying [left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-prop): -*x*.  If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is absolute, the right property specifies the subcontrol's right edge in relation to the parent's right edge (see also [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop)).  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button { right: 2px }  See also [left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-prop), [top](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "top-prop), and [bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-prop). |
| **selection-background-color\*** | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The background of selected text or items.  This property is supported by all widgets that respect the [QWidget::palette](https://doc.qt.io/qt-5/qwidget.html" \l "palette-prop) and that show selection text.  If this property is not set, the default value is whatever is set for the palette's [Highlight](https://doc.qt.io/qt-5/qpalette.html" \l "ColorRole-enum) role.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { selection-background-color: darkblue }  See also [selection-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-color-prop) and [background](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "background-prop). |
| **selection-color\*** | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | The foreground of selected text or items.  This property is supported by all widgets that respect the [QWidget::palette](https://doc.qt.io/qt-5/qwidget.html" \l "palette-prop) and that show selection text.  If this property is not set, the default value is whatever is set for the palette's [HighlightedText](https://doc.qt.io/qt-5/qpalette.html" \l "ColorRole-enum) role.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { selection-color: white }  See also [selection-background-color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "selection-background-color-prop) and [color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color-prop). |
| **show-decoration-selected\*** | [Boolean](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "boolean) | Controls whether selections in a [QListView](https://doc.qt.io/qt-5/qlistview.html) cover the entire row or just the extent of the text.  If this property is not specified, it defaults to the value specified by the current style for the [SH\_ItemView\_ShowDecorationSelected](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  Example:  \* { show-decoration-selected: 1 } |
| **spacing\*** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | Internal spacing in the widget.  This property is supported by [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), checkable [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html)es, [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), and [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html).  If this property is not specified, the default value depends on the widget and on the current style.  Example:  [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html) { spacing: 10 }  See also [padding](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "padding-prop) and [margin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "margin-prop). |
| **subcontrol-origin\*** | [Origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "origin) | The origin rectangle of the subcontrol within the parent element.  If this property is not specified, the default is padding.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::up-button {  image: url(:/images/spinup.png);  subcontrol-origin: content;  subcontrol-position: right top;  }  See also [subcontrol-position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-position-prop). |
| **subcontrol-position\*** | [Alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) | The alignment of the subcontrol within the origin rectangle specified by [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop).  If this property is not specified, it defaults to a value that depends on the subcontrol.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::down-button {  image: url(:/images/spindown.png);  subcontrol-origin: padding;  subcontrol-position: right bottom;  }  See also [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop). |
| **titlebar-show-tooltips-on-buttons**} | bool | Whether tool tips are shown on window title bar buttons. |
| **widget-animation-duration\*** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) | How much an animation should last (in milliseconds). A value equal to zero means that the animations will be disabled.  If this property is not specified, it defaults to the value specified by the current style for the [SH\_Widget\_Animation\_Duration](https://doc.qt.io/qt-5/qstyle.html" \l "StyleHint-enum) style hint.  **This property was added in Qt 5.10.**  Example:  \* { widget-animation-duration: 100 } |
| **text-align** | [Alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) | The alignment of text and icon within the contents of the widget.  If this value is not specified, it defaults to the value that depends on the native style.  Example:  [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) {  text-align: left;  }  This property is currently supported only by [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) and [QProgressBar](https://doc.qt.io/qt-5/qprogressbar.html). |
| **text-decoration** | none underline overline line-through | Additional text effects |
| **top** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is relative (the default), moves a subcontrol by a certain offset down.  If [position](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "position-prop) is absolute, the top property specifies the subcontrol's top edge in relation to the parent's top edge (see also [subcontrol-origin](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "subcontrol-origin-prop)).  If this property is not specified, it defaults to 0.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::up-button { top: 2px }  See also [left](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "left-prop), [right](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "right-prop), and [bottom](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "bottom-prop). |
| **width** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The width of a subcontrol (or a widget in some cases).  If this property is not specified, it defaults to a value that depends on the subcontrol/widget and on the current style.  **Warning:** Unless otherwise specified, this property has no effect when set on widgets. If you want a widget with a fixed width, set the [min-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "min-width-prop) and [max-width](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "max-width-prop) to the same value.  Example:  [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html)::up-button { width: 12px }  See also [height](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "height-prop). |
| **-qt-background-role** | [PaletteRole](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "paletterole) | The background-color for the subcontrol or widget based on the chosen role. |
| **-qt-style-features** | list | The list of CSS properties that you want to apply Qt-specific styles on.  **Note:**The list can only include properties that are not pixmap-based. |

## List of Icons

Icons used in Qt can be customized using the following properties. Each of the properties listed in this section have the type [Icon](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "icon).

Note that for icons to appear in buttons in a [QDialogButtonBox](https://doc.qt.io/qt-5/qdialogbuttonbox.html), you need to set the dialogbuttonbox-buttons-have-icons property to true. Also, to customize the size of the icons, use the icon-size property.

| Name | [QStyle::StandardPixmap](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| --- | --- |
| backward-icon | [QStyle::SP\_ArrowBack](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| cd-icon | [QStyle::SP\_DriveCDIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| computer-icon | [QStyle::SP\_ComputerIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| desktop-icon | [QStyle::SP\_DesktopIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-apply-icon | [QStyle::SP\_DialogApplyButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-cancel-icon | [QStyle::SP\_DialogCancelButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-close-icon | [QStyle::SP\_DialogCloseButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-discard-icon | [QStyle::SP\_DialogDiscardButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-help-icon | [QStyle::SP\_DialogHelpButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-no-icon | [QStyle::SP\_DialogNoButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-ok-icon | [QStyle::SP\_DialogOkButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-open-icon | [QStyle::SP\_DialogOpenButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-reset-icon | [QStyle::SP\_DialogResetButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-save-icon | [QStyle::SP\_DialogSaveButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dialog-yes-icon | [QStyle::SP\_DialogYesButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| directory-closed-icon | [QStyle::SP\_DirClosedIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| directory-icon | [QStyle::SP\_DirIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| directory-link-icon | [QStyle::SP\_DirLinkIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| directory-open-icon | [QStyle::SP\_DirOpenIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dockwidget-close-icon | [QStyle::SP\_DockWidgetCloseButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| downarrow-icon | [QStyle::SP\_ArrowDown](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| dvd-icon | [QStyle::SP\_DriveDVDIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| file-icon | [QStyle::SP\_FileIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| file-link-icon | [QStyle::SP\_FileLinkIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-contentsview-icon | [QStyle::SP\_FileDialogContentsView](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-detailedview-icon | [QStyle::SP\_FileDialogDetailedView](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-end-icon | [QStyle::SP\_FileDialogEnd](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-infoview-icon | [QStyle::SP\_FileDialogInfoView](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-listview-icon | [QStyle::SP\_FileDialogListView](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-new-directory-icon | [QStyle::SP\_FileDialogNewFolder](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-parent-directory-icon | [QStyle::SP\_FileDialogToParent](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| filedialog-start-icon | [QStyle::SP\_FileDialogStart](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| floppy-icon | [QStyle::SP\_DriveFDIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| forward-icon | [QStyle::SP\_ArrowForward](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| harddisk-icon | [QStyle::SP\_DriveHDIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| home-icon | [QStyle::SP\_DirHomeIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| leftarrow-icon | [QStyle::SP\_ArrowLeft](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| messagebox-critical-icon | [QStyle::SP\_MessageBoxCritical](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| messagebox-information-icon | [QStyle::SP\_MessageBoxInformation](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| messagebox-question-icon | [QStyle::SP\_MessageBoxQuestion](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| messagebox-warning-icon | [QStyle::SP\_MessageBoxWarning](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| network-icon | [QStyle::SP\_DriveNetIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| rightarrow-icon | [QStyle::SP\_ArrowRight](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-contexthelp-icon | [QStyle::SP\_TitleBarContextHelpButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-maximize-icon | [QStyle::SP\_TitleBarMaxButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-menu-icon | [QStyle::SP\_TitleBarMenuButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-minimize-icon | [QStyle::SP\_TitleBarMinButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-normal-icon | [QStyle::SP\_TitleBarNormalButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-shade-icon | [QStyle::SP\_TitleBarShadeButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| titlebar-unshade-icon | [QStyle::SP\_TitleBarUnshadeButton](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| trash-icon | [QStyle::SP\_TrashIcon](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |
| uparrow-icon | [QStyle::SP\_ArrowUp](https://doc.qt.io/qt-5/qstyle.html" \l "StandardPixmap-enum) |

## List of Property Types

The following table summarizes the syntax and meaning of the different property types.

| Type | Syntax | Description |
| --- | --- | --- |
| **Alignment** | { top | bottom | left | right | center }\* | Horizontal and/or vertical alignment.  Example:  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) { background-position: bottom center } |
| **Attachment** | { scroll | fixed }\* | Scroll or fixed attachment. |
| **Background** | { [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) | [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url) | [Repeat](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "repeat) | [Alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment) }\* | A sequence of [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush), [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url), [Repeat](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "repeat), and [Alignment](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "alignment). |
| **Boolean** | 0 | 1 | True (1) or false (0).  Example:  [QDialog](https://doc.qt.io/qt-5/qdialog.html) { etch-disabled-text: 1 } |
| **Border** | { [Border Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "border-style) | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush) }\* | Shorthand border property. |
| **Border Image** | none | [Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url) [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number){4} (stretch | repeat){0,2} | A border image is an image that is composed of nine parts (top left, top center, top right, center left, center, center right, bottom left, bottom center, and bottom right). When a border of a certain size is required, the corner parts are used as is, and the top, right, bottom, and left parts are stretched or repeated to produce a border with the desired size.  See the [CSS3 Draft Specification](http://www.w3.org/TR/css3-background/" \l "the-border-image) for details. |
| **Border Style** | dashed | dot-dash | dot-dot-dash | dotted | double | groove | inset | outset | ridge | solid | none | Specifies the pattern used to draw a border. See the [CSS3 Draft Specification](http://www.w3.org/TR/css3-background/" \l "border-style) for details. |
| **Box Colors** | [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush){1,4} | One to four occurrences of [Brush](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "brush), specifying the top, right, bottom, and left edges of a box, respectively. If the left color is not specified, it is taken to be the same as the right color. If the bottom color is not specified, it is taken to be the same as the top color. If the right color is not specified, it is taken to be the same as the top color.  Example:  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: red } /\* red red red red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: red blue } /\* red blue red blue \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: red blue green } /\* red blue green blue \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: red blue green yellow }  /\* red blue green yellow \*/ |
| **Box Lengths** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length){1,4} | One to four occurrences of [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length), specifying the top, right, bottom, and left edges of a box, respectively. If the left length is not specified, it is taken to be the same as the right length. If the bottom length is not specified, is it taken to be the same as the top length. If the right length is not specified, it is taken to be the same as the top length.  Examples:  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-width: 1px } /\* 1px 1px 1px 1px \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-width: 1px 2px } /\* 1px 2px 1px 2px \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-width: 1px 2px 3px } /\* 1px 2px 3px 2px \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-width: 1px 2px 3px 4px } /\* 1px 2px 3px 4px \*/ |
| **Brush** | [Color](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "color) | [Gradient](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "gradient) | [PaletteRole](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "paletterole) | Specifies a Color or a Gradient or an entry in the Palette. |
| **Color** | rgb(*r*, *g*, *b*) | rgba(*r*, *g*, *b*, *a*) | hsv(*h*, *s*, *v*) | hsva(*h*, *s*, *v*, *a*) | hsl(*h*, *s*, *l*) | hsla(*h*, *s*, *l*, *a*) | #*rrggbb* | [Color Name](https://doc.qt.io/qt-5/qcolor.html" \l "setNamedColor) | Specifies a color as RGB (red, green, blue), RGBA (red, green, blue, alpha), HSV (hue, saturation, value), HSVA (hue, saturation, value, alpha), HSL (hue, saturation, lightness), HSLA (hue, saturation, lightness, alpha) or a named color. The rgb() or rgba() syntax can be used with integer values between 0 and 255, or with percentages. The value of s, v, l and a in hsv(), hsva() hsl() or hsla() must all be in the range 0-255 or with percentages, the value of h must be in the range 0-359. The support for HSL(A) is available since 5.13.  Examples:  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: red } /\* opaque red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: #FF0000 } /\* opaque red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: rgba(255, 0, 0, 75%) } /\* 75% opaque red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: rgb(255, 0, 0) } /\* opaque red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: rgb(100%, 0%, 0%) } /\* opaque red \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: hsv(60, 100%, 100%) } /\* opaque yellow \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: hsva(240, 255, 255, 75%) } /\* 75% blue \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: hsl(60, 100%, 50%) } /\* opaque yellow \*/  [QLabel](https://doc.qt.io/qt-5/qlabel.html) { border-color: hsla(240, 255, 50%, 75%) } /\* 75% blue \*/  **Note:**The RGB colors allowed are the same as those allowed with CSS 2.1, as listed [here](http://www.w3.org/TR/CSS21/syndata.html" \l "color-units). |
| **Font** | ([Font Style](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-style) | [Font Weight](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-weight)){0,2} [Font Size](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "font-size) String | Shorthand font property. |
| **Font Size** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length) | The size of a font. |
| **Font Style** | normal | italic | oblique | The style of a font. |
| **Font Weight** | normal | bold | 100 | 200 ... | 900 | The weight of a font. |
| **Gradient** | qlineargradient | qradialgradient | qconicalgradient | Specifies gradient fills. There are three types of gradient fills:   * *Linear* gradients interpolate colors between start and end points. * *Radial* gradients interpolate colors between a focal point and end points on a circle surrounding it. * *Conical* gradients interpolate colors around a center point.   Gradients are specified in Object Bounding Mode. Imagine the box in which the gradient is rendered, to have its top left corner at (0, 0) and its bottom right corner at (1, 1). Gradient parameters are then specified as percentages from 0 to 1. These values are extrapolated to actual box coordinates at runtime. It is possible specify values that lie outside the bounding box (-0.6 or 1.8, for instance).  **Warning:** The stops have to appear sorted in ascending order.  Examples:  /\* linear gradient from white to green \*/  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) {  background: qlineargradient(x1:0, y1:0, x2:1, y2:1,  stop:0 white, stop: 0.4 gray, stop:1 green)  }  /\* linear gradient from white to green \*/  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) {  background: qlineargradient(x1:0, y1:0, x2:1, y2:1,  stop:0 white, stop: 0.4 rgba(10, 20, 30, 40),  stop:1 rgb(0, 200, 230, 200))  }  /\* conical gradient from white to green \*/  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) {  background: qconicalgradient(cx:0.5, cy:0.5, angle:30,  stop:0 white, stop:1 #00FF00)  }  /\* radial gradient from white to green \*/  [QTextEdit](https://doc.qt.io/qt-5/qtextedit.html) {  background: qradialgradient(cx:0, cy:0, radius: 1,  fx:0.5, fy:0.5, stop:0 white, stop:1 green)  } |
| **Icon** | ([Url](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "url) (disabled | active | normal | selected)? (on | off)? )\* | A list of url, [QIcon::Mode](https://doc.qt.io/qt-5/qicon.html" \l "Mode-enum) and [QIcon::State](https://doc.qt.io/qt-5/qicon.html" \l "State-enum).  Example:  \* {  file-icon: url(file.png),  url(file\_selected.png) selected;  }  [QMessageBox](https://doc.qt.io/qt-5/qmessagebox.html) {  dialogbuttonbox-buttons-have-icons: true;  dialog-ok-icon: url(ok.svg);  dialog-cancel-icon: url(cancel.png),  url(grayed\_cancel.png) disabled;  } |
| **Length** | [Number](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "number) (px | pt | em | ex)? | A number followed by a measurement unit. The CSS standard recommends that user agents must [ignore](http://www.w3.org/TR/CSS21/syndata.html" \l "illegalvalues) a declaration with an illegal value. In Qt, it is mandatory to specify measurement units. For compatibility with earlier versions of Qt, numbers without measurement units are treated as pixels in most contexts. The supported units are:   * px: pixels * pt: the size of one point (i.e., 1/72 of an inch) * em: the em width of the font (i.e., the width of 'M') * ex: the ex width of the font (i.e., the height of 'x')   However, Qt is limited to font sizes in pt and px and any other size must be in px, em or ex. |
| **Number** | A decimal integer or a real number | Examples: 0, 18, +127, -255, 12.34, -.5, 0009. |
| **Origin** | margin | border | padding | content | Indicates which of four rectangles to use.   * margin: The margin rectangle. The margin falls outside the border. * border: The border rectangle. This is where any border is drawn. * padding: The padding rectangle. Unlike the margins, padding is located inside the border. * content: The content rectangle. This specifies where the actual contents go, excluding any padding, border, or margin.   See also [The Box Model](https://doc.qt.io/qt-5/stylesheet-customizing.html" \l "the-box-model). |
| **[PaletteRole](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "paletterole)** | alternate-base | base | bright-text | button | button-text | dark | highlight | highlighted-text | light | link | link-visited | mid | midlight | shadow | text | window | window-text | These values correspond the [Color roles](https://doc.qt.io/qt-5/qpalette.html" \l "ColorRole-enum) in the widget's [QPalette](https://doc.qt.io/qt-5/qpalette.html).  For example,  [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) { color: palette(dark); } |
| **Radius** | [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length){1, 2} | One or two occurrences of [Length](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "length). If only one length is specified, it is used as the radius of the quarter circle defining the corner. If two lengths are specified, the first length is the horizontal radius of a quarter ellipse, whereas the second length is the vertical radius. |
| **Repeat** | repeat-x | repeat-y | repeat | no-repeat | A value indicating the nature of repetition.   * repeat-x: Repeat horizontally. * repeat-y: Repeat vertically. * repeat: Repeat horizontally and vertically. * no-repeat: Don't repeat. |
| **Url** | url(*filename*) | *filename* is the name of a file on the local disk or stored using [The Qt Resource System](https://doc.qt.io/qt-5/resources.html). Setting an image implicitly sets the width and height of the element. |

## List of Pseudo-States

The following pseudo-states are supported:

| Pseudo-State | Description |
| --- | --- |
| :active | This state is set when the widget resides in an active window. |
| :adjoins-item | This state is set when the [::branch](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "branch-sub) of a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) is adjacent to an item. |
| :alternate | This state is set for every alternate row whe painting the row of a [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) when [QAbstractItemView::alternatingRowColors](https://doc.qt.io/qt-5/qabstractitemview.html" \l "alternatingRowColors-prop)() is set to true. |
| :bottom | The item is positioned at the bottom. For example, a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) that has its tabs positioned at the bottom. |
| :checked | The item is checked. For example, the [checked](https://doc.qt.io/qt-5/qabstractbutton.html" \l "checked-prop) state of [QAbstractButton](https://doc.qt.io/qt-5/qabstractbutton.html). |
| :closable | The items can be closed. For example, the [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) has the [QDockWidget::DockWidgetClosable](https://doc.qt.io/qt-5/qdockwidget.html" \l "DockWidgetFeature-enum) feature turned on. |
| :closed | The item is in the closed state. For example, an non-expanded item in a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) |
| :default | The item is the default. For example, a [default](https://doc.qt.io/qt-5/qpushbutton.html" \l "default-prop) [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) or a default action in a [QMenu](https://doc.qt.io/qt-5/qmenu.html). |
| :disabled | The item is [disabled](https://doc.qt.io/qt-5/qwidget.html" \l "enabled-prop). |
| :editable | The [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) is editable. |
| :edit-focus | The item has edit focus (See [QStyle::State\_HasEditFocus](https://doc.qt.io/qt-5/qstyle.html" \l "StateFlag-enum)). This state is available only for Qt Extended applications. |
| :enabled | The item is [enabled](https://doc.qt.io/qt-5/qwidget.html" \l "enabled-prop). |
| :exclusive | The item is part of an exclusive item group. For example, a menu item in a exclusive [QActionGroup](https://doc.qt.io/qt-5/qactiongroup.html). |
| :first | The item is the first (in a list). For example, the first tab in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| :flat | The item is flat. For example, a [flat](https://doc.qt.io/qt-5/qpushbutton.html" \l "flat-prop) [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html). |
| :floatable | The items can be floated. For example, the [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) has the [QDockWidget::DockWidgetFloatable](https://doc.qt.io/qt-5/qdockwidget.html" \l "DockWidgetFeature-enum) feature turned on. |
| :focus | The item has [input focus](https://doc.qt.io/qt-5/qwidget.html" \l "focus-prop). |
| :has-children | The item has children. For example, an item in a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) that has child items. |
| :has-siblings | The item has siblings. For example, an item in a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html) that siblings. |
| :horizontal | The item has horizontal orientation |
| :hover | The mouse is hovering over the item. |
| :indeterminate | The item has indeterminate state. For example, a [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html) or [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html) is [partially checked](https://doc.qt.io/qt-5/qt.html" \l "CheckState-enum). |
| :last | The item is the last (in a list). For example, the last tab in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| :left | The item is positioned at the left. For example, a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) that has its tabs positioned at the left. |
| :maximized | The item is maximized. For example, a maximized [QMdiSubWindow](https://doc.qt.io/qt-5/qmdisubwindow.html). |
| :middle | The item is in the middle (in a list). For example, a tab that is not in the beginning or the end in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| :minimized | The item is minimized. For example, a minimized [QMdiSubWindow](https://doc.qt.io/qt-5/qmdisubwindow.html). |
| :movable | The item can be moved around. For example, the [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) has the [QDockWidget::DockWidgetMovable](https://doc.qt.io/qt-5/qdockwidget.html" \l "DockWidgetFeature-enum) feature turned on. |
| :no-frame | The item has no frame. For example, a frameless [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html) or [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html). |
| :non-exclusive | The item is part of a non-exclusive item group. For example, a menu item in a non-exclusive [QActionGroup](https://doc.qt.io/qt-5/qactiongroup.html). |
| :off | For items that can be toggled, this applies to items in the "off" state. |
| :on | For items that can be toggled, this applies to widgets in the "on" state. |
| :only-one | The item is the only one (in a list). For example, a lone tab in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| :open | The item is in the open state. For example, an expanded item in a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html), or a [QComboBox](https://doc.qt.io/qt-5/qcombobox.html) or [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html) with an open menu. |
| :next-selected | The next item (in a list) is selected. For example, the selected tab of a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) is next to this item. |
| :pressed | The item is being pressed using the mouse. |
| :previous-selected | The previous item (in a list) is selected. For example, a tab in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) that is next to the selected tab. |
| :read-only | The item is marked read only or non-editable. For example, a read only [QLineEdit](https://doc.qt.io/qt-5/qlineedit.html) or a non-editable [QComboBox](https://doc.qt.io/qt-5/qcombobox.html). |
| :right | The item is positioned at the right. For example, a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) that has its tabs positioned at the right. |
| :selected | The item is selected. For example, the selected tab in a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) or the selected item in a [QMenu](https://doc.qt.io/qt-5/qmenu.html). |
| :top | The item is positioned at the top. For example, a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) that has its tabs positioned at the top. |
| :unchecked | The item is [unchecked](https://doc.qt.io/qt-5/qabstractbutton.html" \l "checked-prop). |
| :vertical | The item has vertical orientation. |
| :window | The widget is a window (i.e top level widget) |

## List of Sub-Controls

The following subcontrols are available:

| Sub-Control | Description |
| --- | --- |
| ::add-line | The button to add a line of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::add-page | The region between the handle (slider) and the [add-line](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "add-line-sub) of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::branch | The branch indicator of a [QTreeView](https://doc.qt.io/qt-5/qtreeview.html). |
| ::chunk | The progress chunk of a [QProgressBar](https://doc.qt.io/qt-5/qprogressbar.html). |
| ::close-button | The close button of a [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) or tabs of [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) |
| ::corner | The corner between two scrollbars in a [QAbstractScrollArea](https://doc.qt.io/qt-5/qabstractscrollarea.html) |
| ::down-arrow | The down arrow of a [QComboBox](https://doc.qt.io/qt-5/qcombobox.html), [QHeaderView](https://doc.qt.io/qt-5/qheaderview.html) (sort indicator), [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html) or [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html). |
| ::down-button | The down button of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html) or a [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html). |
| ::drop-down | The drop-down button of a [QComboBox](https://doc.qt.io/qt-5/qcombobox.html). |
| ::float-button | The float button of a [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html) |
| ::groove | The groove of a [QSlider](https://doc.qt.io/qt-5/qslider.html). |
| ::indicator | The indicator of a [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html), a [QCheckBox](https://doc.qt.io/qt-5/qcheckbox.html), a [QRadioButton](https://doc.qt.io/qt-5/qradiobutton.html), a checkable [QMenu](https://doc.qt.io/qt-5/qmenu.html) item or a checkable [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html). |
| ::handle | The handle (slider) of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html), a [QSplitter](https://doc.qt.io/qt-5/qsplitter.html), or a [QSlider](https://doc.qt.io/qt-5/qslider.html). |
| ::icon | The icon of a [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html) or a [QMenu](https://doc.qt.io/qt-5/qmenu.html). |
| ::item | An item of a [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html), a [QMenuBar](https://doc.qt.io/qt-5/qmenubar.html), a [QMenu](https://doc.qt.io/qt-5/qmenu.html), or a [QStatusBar](https://doc.qt.io/qt-5/qstatusbar.html). |
| ::left-arrow | The left arrow of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::left-corner | The left corner of a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). For example, this control can be used to control position the left corner widget in a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). |
| ::menu-arrow | The arrow of a [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html) with a menu. |
| ::menu-button | The menu button of a [QToolButton](https://doc.qt.io/qt-5/qtoolbutton.html). |
| ::menu-indicator | The menu indicator of a [QPushButton](https://doc.qt.io/qt-5/qpushbutton.html). |
| ::right-arrow | The right arrow of a [QMenu](https://doc.qt.io/qt-5/qmenu.html) or a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::pane | The pane (frame) of a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). |
| ::right-corner | The right corner of a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). For example, this control can be used to control the position the right corner widget in a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). |
| ::scroller | The scroller of a [QMenu](https://doc.qt.io/qt-5/qmenu.html) or [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| ::section | The section of a [QHeaderView](https://doc.qt.io/qt-5/qheaderview.html). |
| ::separator | The separator of a [QMenu](https://doc.qt.io/qt-5/qmenu.html) or in a [QMainWindow](https://doc.qt.io/qt-5/qmainwindow.html). |
| ::sub-line | The button to subtract a line of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::sub-page | The region between the handle (slider) and the [sub-line](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "sub-line-sub) of a [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html). |
| ::tab | The tab of a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) or [QToolBox](https://doc.qt.io/qt-5/qtoolbox.html). |
| ::tab-bar | The tab bar of a [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). This subcontrol exists only to control the position of the [QTabBar](https://doc.qt.io/qt-5/qtabbar.html) inside the [QTabWidget](https://doc.qt.io/qt-5/qtabwidget.html). To style the tabs using the [::tab](https://doc.qt.io/qt-5/stylesheet-reference.html" \l "tab-sub) subcontrol. |
| ::tear | The tear indicator of a [QTabBar](https://doc.qt.io/qt-5/qtabbar.html). |
| ::tearoff | The tear-off indicator of a [QMenu](https://doc.qt.io/qt-5/qmenu.html). |
| ::text | The text of a [QAbstractItemView](https://doc.qt.io/qt-5/qabstractitemview.html). |
| ::title | The title of a [QGroupBox](https://doc.qt.io/qt-5/qgroupbox.html) or a [QDockWidget](https://doc.qt.io/qt-5/qdockwidget.html). |
| ::up-arrow | The up arrow of a [QHeaderView](https://doc.qt.io/qt-5/qheaderview.html) (sort indicator), [QScrollBar](https://doc.qt.io/qt-5/qscrollbar.html) or a [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html). |
| ::up-button | The up button of a [QSpinBox](https://doc.qt.io/qt-5/qspinbox.html). |

See [Customizing the QPushButton's Menu Indicator Sub-Control](https://doc.qt.io/qt-5/stylesheet-examples.html" \l "customizing-the-qpushbutton-s-menu-indicator-sub-control) for an example of how to customize a subcontrol.

[Customizing Qt Widgets Using Style Sheets](https://doc.qt.io/qt-5/stylesheet-customizing.html)[Qt Style Sheets Examples](https://doc.qt.io/qt-5/stylesheet-examples.html)