# Qt 之 QProgressBar

# 简述

QProgressBar部件提供了一个水平或垂直进度条。

进度条用于给用户操作一个进度指示，并向它们说明应用程序仍在运行。

# 详细描述

可以通过setRange()来设置进度的最小值和最大值（取值范围），也可使用setMinimum()和setMaximum()来单独设定；成员函数setValue()用于设置当前的运行值；调用reset()则会让进度条重新回到开始。

当前值设置完成以后，将显示已完成的百分比，计算百分比的公式为：(value() - minimum()) / (maximum() - minimum())。

如果最小值和最大值都设置为0，进度条会显示一个繁忙指示，而不会显示当前值。有时候这很有用，例如：当使用QNetworkAccessManager下载东西，无法确定被下载项大小时。

可以通过setOrientation()指定进度条的方向 - 水平/垂直。

此外，成员函数setInvertedAppearance()用于设置进度条的行进方向，如果参数为true，可将进度方向设置为默认方向的反方向。

如果不需要显示进度条上的文本，可以使用setTextVisible()来隐藏。

# 读取方向

枚举 QProgressBar::Direction

指定垂直进度条文本的读取方向。

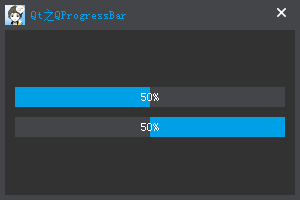
| **常量** | **值** | **描述** |
| --- | --- | --- |
| QProgressBar::TopToBottom | 0 | 文本是顺时针旋转了90度 |
| QProgressBar::BottomToTop | 1 | 文本是逆时针旋转90度 |

这个属性对水平进度条没有影响。默认情况下，读取方向为：QProgressBar::TopToBottom。

# 进度方向

当水平显示进度时，可以从左到右，也可以从右到左；同样，垂直显示进度时，可以从上到下，也可以从下到上。

## 效果



## 源码

QProgressBar \*pProgressBar = new QProgressBar(this);

pProgressBar->setOrientation(Qt::Horizontal); *// 水平方向*

pProgressBar->setMinimum(0); *// 最小值*

pProgressBar->setMaximum(100); *// 最大值*

pProgressBar->setValue(50); *// 当前进度*

QProgressBar \*pProgressBar2 = new QProgressBar(this);

pProgressBar2->setOrientation(Qt::Horizontal); *// 水平方向*

pProgressBar2->setMinimum(0); *// 最小值*

pProgressBar2->setMaximum(100); *// 最大值*

pProgressBar2->setValue(50); *// 当前进度*

pProgressBar2->setInvertedAppearance(true); *// 反方向*

垂直方向只需要将Qt::Horizontal替换为Qt::Vertical即可，这里就不再演示了。

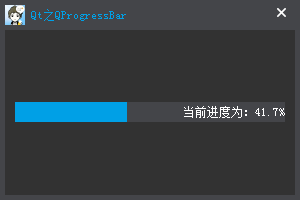
# 文本显示

setFormat()用于生成当前文本字符串。

* %p - 被完成百分比所取代
* %v - 被当前值所取代
* %m - 被总步数所取代

默认值是 “%p%”。

## 效果



## 源码

QProgressBar \*pProgressBar = new QProgressBar(this);

pProgressBar->setOrientation(Qt::Horizontal); *// 水平方向*

pProgressBar->setMinimum(0); *// 最小值*

pProgressBar->setMaximum(4800); *// 最大值*

pProgressBar->setValue(2000); *// 当前进度*

double dProgress = (pProgressBar->value() - pProgressBar->minimum()) \* 100.0

/ (pProgressBar->maximum() - pProgressBar->minimum());

pProgressBar->setFormat(QString::fromLocal8Bit("当前进度为：%1%").arg(QString::number(dProgress, 'f', 1)));

pProgressBar->setAlignment(Qt::AlignRight | Qt::AlignVCenter); *// 对齐方式*

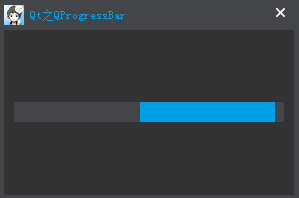
如果要显示百分比，可以直接使用”%p%”（比如：41%），但是如果我们要精确显示（比如：41.7%），就得自己计算了，进度公式参考前面。

通过setAlignment()，可以指定显示文本的对齐方式（也可通过QSS样式中的属性text-align来指定）。

# 繁忙指示

如果最小值和最大值都设置为0，进度条会显示了一个繁忙指示，而不会显示当前的值。

## 效果



## 源码

QProgressBar \*pProgressBar = new QProgressBar(this);

pProgressBar->setOrientation(Qt::Horizontal); *// 水平方向*

pProgressBar->setMinimum(0); *// 最小值*

pProgressBar->setMaximum(0); *// 最大值*

# 

# QSS

QProgressBar{

border: none;

color: white;

text-align: center;

background: rgb(68, 69, 73);

}

QProgressBar::chunk {

border: none;

background: rgb(0, 160, 230);

}

进度条很常用，进度对话框QProgressDialog就是采用的QProgressBar。在我们日常生活中其实经常会遇到，例如：Windows下拷贝/删除文件。