# Qt 之水平/垂直布局（QBoxLayout、QHBoxLayout、QVBoxLayout）

# 简述

QBoxLayout可以在水平方向或垂直方向上排列控件，由QHBoxLayout、QVBoxLayout所继承。

QHBoxLayout：水平布局，在水平方向上排列控件，即：左右排列。   
QVBoxLayout：垂直布局，在垂直方向上排列控件，即：上下排列。

通过查看源码，我们可以发现，水平布局、垂直布局除了构造时的方向（LeftToRight、TopToBottom）不同外，其它均相同。

下面我们以QHBoxLayout为例，来讲解QBoxLayout的常用功能。

# 使用

下面介绍下水平布局的基本使用。

## 效果

这里写图片描述

## 源码

下面，我们创建5个按钮，然后添加至水平不居中。

QPushButton \*pButton1 = new QPushButton("One");

QPushButton \*pButton2 = new QPushButton("Two");

QPushButton \*pButton3 = new QPushButton("Three");

QPushButton \*pButton4 = new QPushButton("Four");

QPushButton \*pButton5 = new QPushButton("Five");

QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addWidget(pButton1);

pHLayout->addWidget(pButton2);

pHLayout->addWidget(pButton3);

pHLayout->addWidget(pButton4);

pHLayout->addWidget(pButton5);

setLayout(pHLayout);

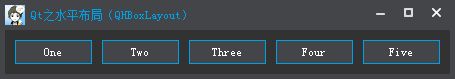
# 常用接口

我们可以看到，默认的外边距为0，为了美观性我们可以设置下Margin。

* setMargin(int)
* setContentsMargins(int left, int top, int right, int bottom);
* setContentsMargins(const QMargins &margins)   
  设置外边距

setMargin可以设置左、上、右、下的外边距，设置之后，他们的外边距是相同的。   
setContentsMargins与其功能相同，但是可以将左、上、右、下的外边距设置为不同的值。

这里我使用setMargin(10)将外边距设置为10。



* setSpacing(int)   
  设置间距

一般情况下，会有一个默认间距值，为了保持所有布局的统一性，或者你需要一个更合适的间距值，则需要手动设置。

这里我使用setSpacing(0)将间距设置为0。

这里写图片描述

* addStretch()   
  添加了一个伸缩空间（QSpacerItem）。

**居右**

这里写图片描述

在第一个控件之前添加伸缩，这样所有的控件就会居右显示。

QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addStretch(); *// 添加伸缩*

pHLayout->addWidget(pButton1);

pHLayout->addWidget(pButton2);

pHLayout->addWidget(pButton3);

pHLayout->addWidget(pButton4);

pHLayout->addWidget(pButton5);

**居左**

这里写图片描述

在最后一个控件之后添加伸缩，这样所有的控件就会居左显示。

QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addWidget(pButton1);

pHLayout->addWidget(pButton2);

pHLayout->addWidget(pButton3);

pHLayout->addWidget(pButton4);

pHLayout->addWidget(pButton5);

pHLayout->addStretch(); *// 添加伸缩*

**居中**

这里写图片描述

在第一个控件之前、最后一个控件之后添加伸缩，这样所有的控件就会居中显示。

QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addStretch(); *// 第一个控件之前添加伸缩*

pHLayout->addWidget(pButton1);

pHLayout->addWidget(pButton2);

pHLayout->addWidget(pButton3);

pHLayout->addWidget(pButton4);

pHLayout->addWidget(pButton5);

pHLayout->addStretch(); *// 最后一个控件之后添加伸缩*

pHLayout->setSpacing(10);

**均分**

这里写图片描述

在每一个控件之间都添加伸缩，这样所有的控件之间的间距都会相同。

QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addStretch();

pHLayout->addWidget(pButton1);

pHLayout->addStretch();

pHLayout->addWidget(pButton2);

pHLayout->addStretch();

pHLayout->addWidget(pButton3);

pHLayout->addStretch();

pHLayout->addWidget(pButton4);

pHLayout->addStretch();

pHLayout->addWidget(pButton5);

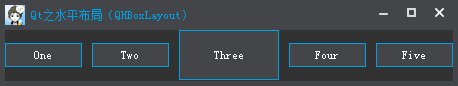
pHLayout->addStretch();

pHLayout->setSpacing(0);

* addWidget(QWidget \*, int stretch = 0, Qt::Alignment alignment = 0)   
  添加控件

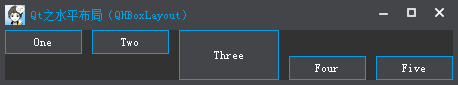
默认的，我们添加控件至水平布局中，默认都是垂直方向居中对齐的。

例如：



其中有控件大小不相同的时候就会看得很明显了，如果我们需要将其中的某些控件居上、居下显示，那么可以使用对齐方式Qt::Alignment。

下面，我们使用向上、向下对齐来设置其它控件。



QHBoxLayout \*pHLayout = new QHBoxLayout();

pHLayout->addStretch();

*// 水平居左 垂直居上*

pHLayout->addWidget(pButton1, 0 , Qt::AlignLeft | Qt::AlignTop);

pHLayout->addWidget(pButton2, 0 , Qt::AlignLeft | Qt::AlignTop);

pHLayout->addWidget(pButton3);

*// 水平居左 垂直居下*

pHLayout->addWidget(pButton4, 0 , Qt::AlignLeft | Qt::AlignBottom);

pHLayout->addWidget(pButton5, 0 , Qt::AlignLeft | Qt::AlignBottom);

pHLayout->setSpacing(10);

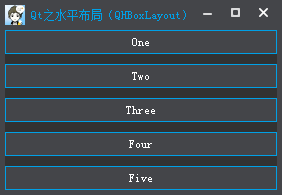
* setDirection(Direction)   
  设置布局方向

可以设置从左到右、从右到左、从上到下、从下到上等。。。

setDirection(QBoxLayout::RightToLeft)

这里写图片描述

setDirection(QBoxLayout::TopToBottom);



既然使用了QHBoxLayout，一般就不建议使用TopToBottom或者BottomToTop，如果实在确定不了方向，或者方向可以随意变化，那么建议使用QBoxLayout。

* setStretchFactor(QWidget \*w, int stretch);
* setStretchFactor(QLayout \*l, int stretch);   
  设置控件、布局的拉伸系数

当窗体大小变化时，控件会根据拉伸系数来做相应的调整。

这里写图片描述

setStretchFactor(pButton1, 1);   
setStretchFactor(pButton2, 2);

设置pButton1的拉伸系数为1，pButton2拉伸系数为2，当窗体变大时，会优先将pButton2进行拉伸，当达到一定程度时，再拉伸pButton1，pButton1与pButton2的宽度比例为1:2。

# 总结

上面介绍了基本所有常用的接口使用，还有一些inset…接口，和它们功能相同，只不过是需要传递控件所在的索引index。常用的这些接口掌握了，其它布局QVBoxLayout、QGridLayout功能也相同或类似，一通百通。