

# qt

布局例子

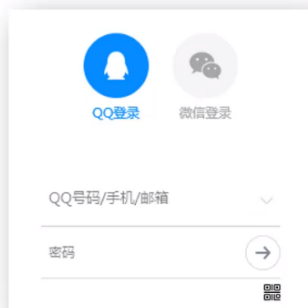
仿照QQ登录界面

四则运算计算器

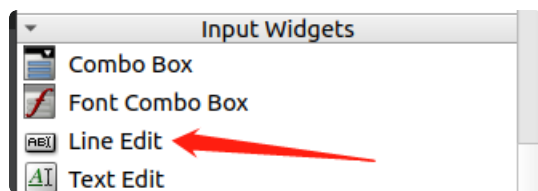
## 布局例子

仿照QQ登录界面

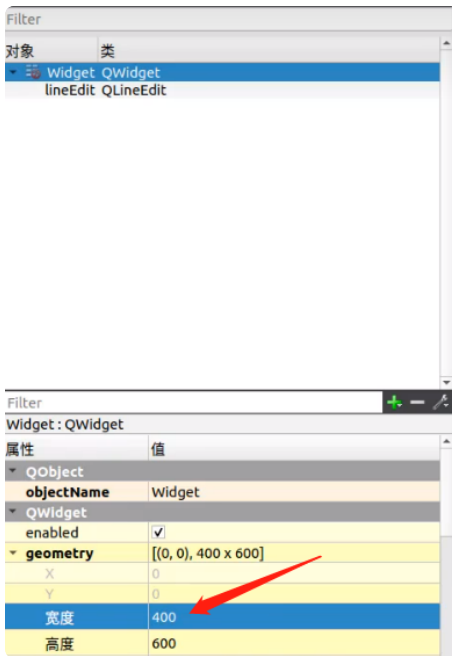
### QLineEdit仿QQ登录界面制作



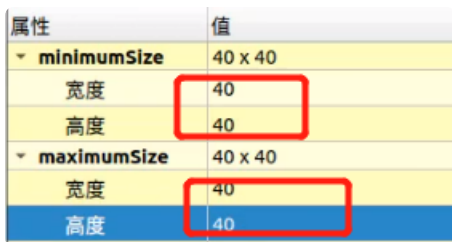
使用控件-单行输入框：



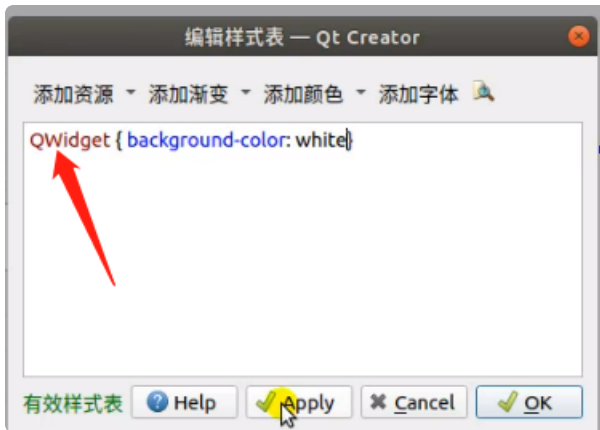
设置面板的宽度,电脑宽800, 手机一般宽400,



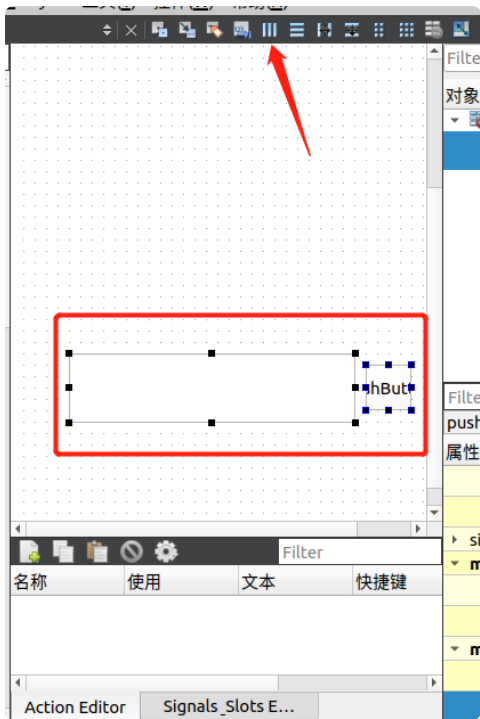
加按钮，固定大小



设置面板背景，注意第二个字母也是大写



将按钮和输入框框选后，水平布局



设置边框背景可以顺便去除边框

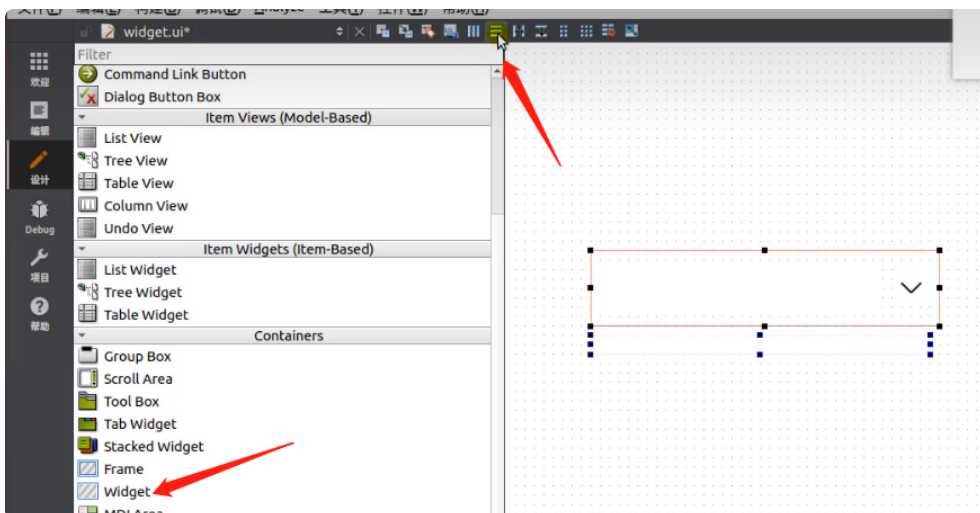
CSS | [复制代码](#)

```

1 QPushButton {
2     border-image: url(/images/rad.png)
3 }

```

横线用Widget 实现，并与上方垂直对齐，设置固定高度2px,设置背景色

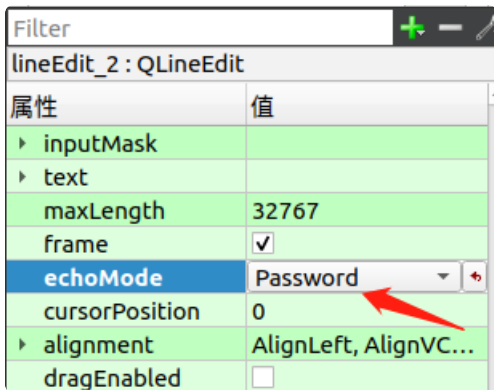


▼ <b>minimumSize</b>	0 x 2
宽度	0
高度	2
▼ <b>maximumSize</b>	16777215 x 2
宽度	16777215
高度	2

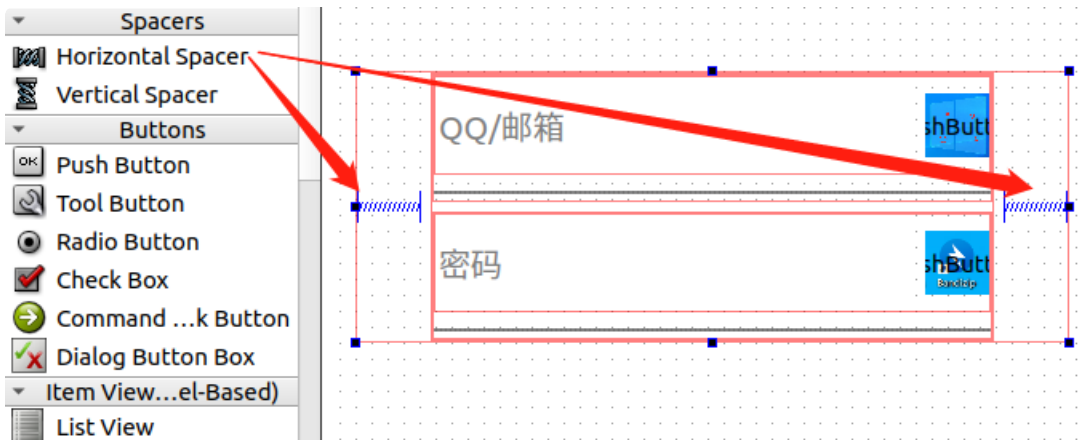
用户名输入框基本完成，密码框类似，直接复制用户的页面，再加上垂直布局，再修改图标，文字等。



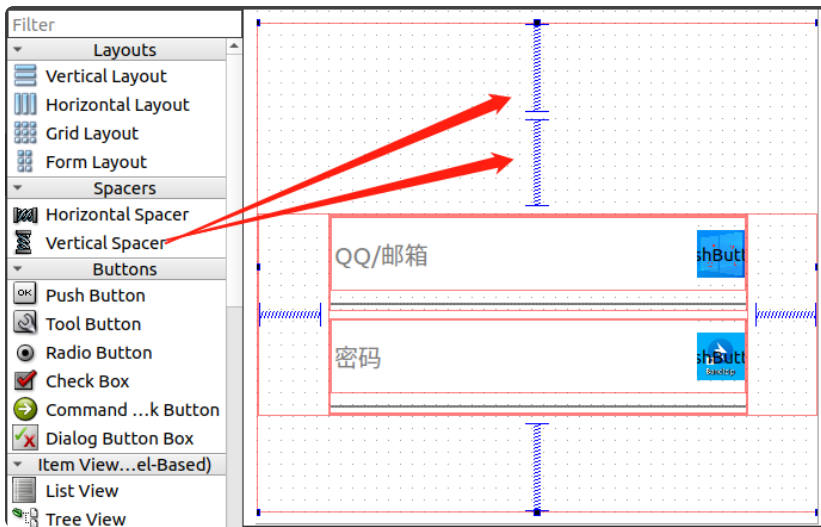
密码隐藏显示



使用2个水平弹簧增加两边的间隔，并使用水平布局，将弹簧与内容水平



同理设置垂直弹簧，上面多放点



因为界面可以被压缩，所以设置 最外层的垂直布局为整体页面的布局

```

widget.cpp
Widget::Widget(QWidget *) -> void

#include "widget.h"
#include "ui_widget.h"

Widget::Widget(QWidget *parent)
    : QWidget(parent)
    , ui(new Ui::Widget)
{
    ui->setupUi(this);
    this->setLayout(ui->verticalLayout_4);
}

Widget::~Widget()
{
    delete ui;
}
    
```

## 四则运算计算器

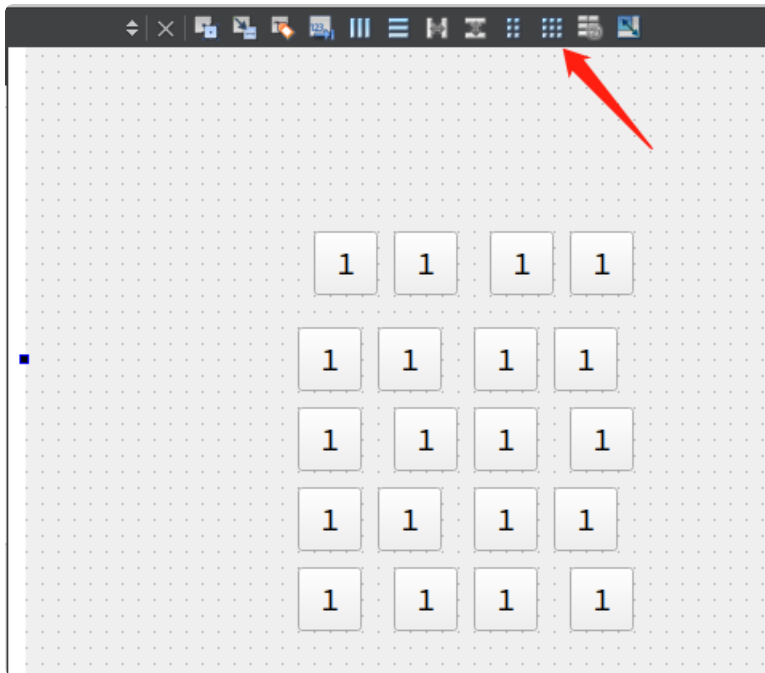
需要使用到 STL 库中的 栈 数据结构，库的名称可能稍有变化

先绘制 UI 页面

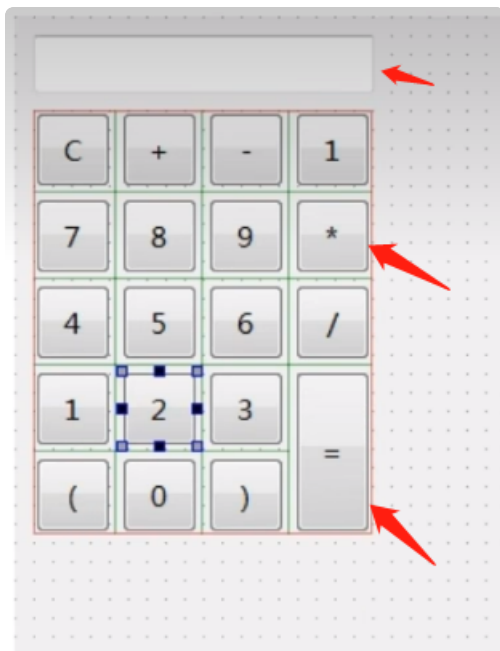
对单个pushbutton设置样式，如字体，固定大小等

▲ <b>minimumSize</b>	40 x 40
宽度	40
高度	40
▲ <b>maximumSize</b>	40 x 40
宽度	40
高度	40

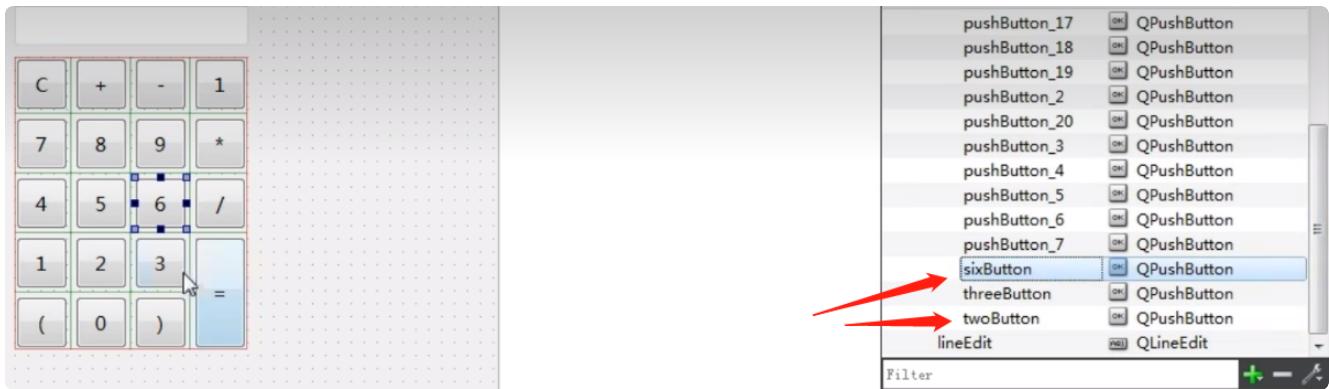
复制多个按钮，并排出基本的界面，再使用栅格布局



对于显示框可使用 Line Edit 部件，等于 符号的高度占两个，设置高度为90即可，对于不好显示的符号如乘除可以使用背景图片。



为了编程需要，可以自定义部件名称。



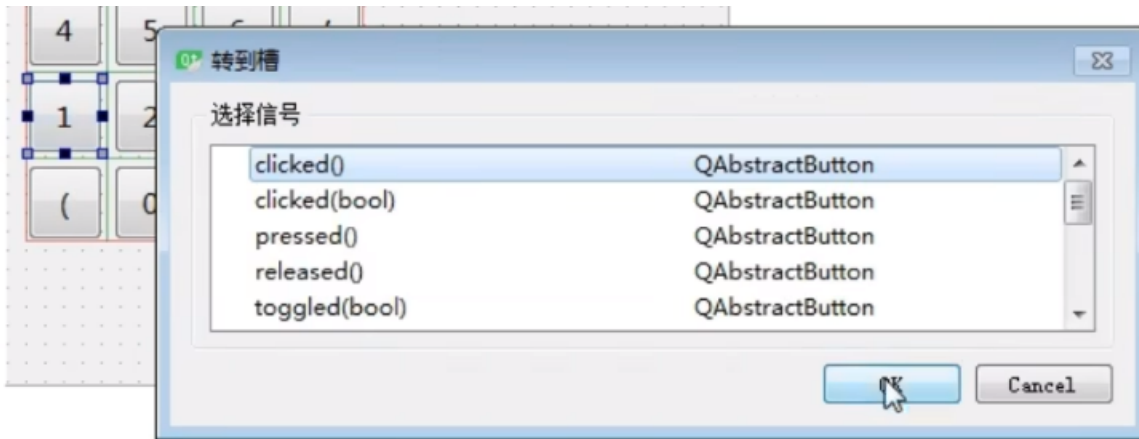
在点击等号之前，需要使用一个字符串记录我们的按键输入值

```

18 private:
19     Ui::Widget *ui;
20     QString expression;
21 };
22
23 #endif // WIDGET_H
24

```

对按钮设置点击信号，并且书写逻辑，此处如 1 按钮



```

void Widget::on_pushButton_9_clicked()
{
    expression += "1";
    ui->lineEdit->setText(expression);
}

```

固定窗口大小，设置标题，字体等

```
widget.cpp Widget::Widget(QWidget *)
{
    QWidget(parent),
    ui(new Ui::Widget)
    {
        ui->setupUi(this);

        this->setMaximumSize(200, 280);
        this->setMinimumSize(200, 280);

        this->setWindowTitle("计算器");

        QFont f("仿宋", 14);    //字体对象
        ui->mainLineEdit->setFont(f);
    }
}
```

将其它按钮都转到槽

```
void Widget::on_pushButton_9_clicked()
{
    expression += "1";
    ui->lineEdit->setText(expression);
}

void Widget::on_pushButton_clicked()
{
    expression += "(";
    ui->lineEdit->setText(expression);
}

void Widget::on_pushButton_6_clicked()
{
    expression += "2";
    ui->lineEdit->setText(expression);
}
```

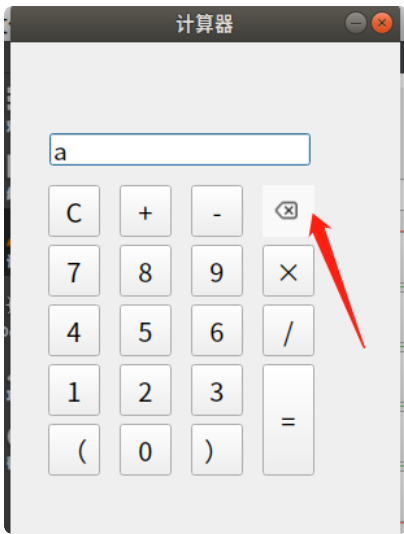
清空按钮的槽函数，字符串也是调用函数，而不是赋值空字符串

```
void Widget::on_pushButton_3_clicked()
{
    expression.clear();
    ui->lineEdit->clear();
}
```

删除按钮的图片如果没有，可以自己截图，设置为icon后，默认大小是16，尽管我们的图片的40，所以我们需改大。



Selected On	
iconSize	40 x 40
宽度	40
高度	40
shortcut	



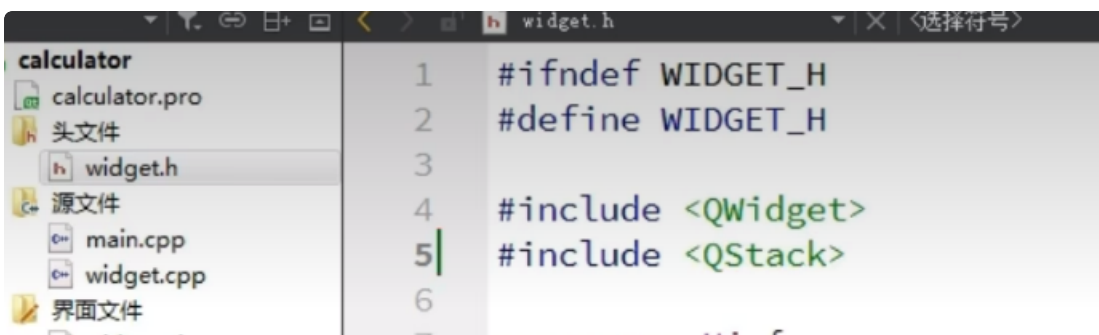
写删除的槽函数，自己在帮助文档里可能没那么快找到需要使用的函数，可以百度。

```
void Widget::on_pushButton_19_clicked()
{
    expression.chop(1); // 字符串末尾删除函数，参数是删除几个
    ui->lineEdit->setText(expression);
}
```

改变等号按钮样式

```
18 ui->pushButton_19->setIcon(icon);
19
20 //改变按钮背景色
21 ui->pushButton_4->setStyleSheet("background: skyblue");
```

使用STL库,将对应的头文件包含即可



最后就是实现等号的槽函数，需要判断很多条件，及运算符的优先级，算法较复杂，略

