1. QMainWindow
   1. 菜单栏 最多有一个
      1. QMenuBar \* bar = MenuBar();
      2. setMenuBar( bar )
      3. QMenu \* fileMenu = bar -> addMenu(“文件”) 创建菜单
      4. QAction \* newAction = fileMenu ->addAction(“新建”); 创建菜单项
      5. 添加分割线 fileMenu->addSeparator();
   2. 工具栏 可以有多个
      1. QToolBar \* toolbar = new QToolBar(this);
      2. addToolBar( 默认停靠区域， toolbar ); Qt::LeftToolBarArea
      3. 设置 后期停靠区域，设置浮动，设置移动
      4. 添加菜单项 或者添加 小控件
   3. 状态栏 最多一个
      1. QStatusBar \* stBar = statusBar();
      2. 设置到窗口中 setStatusBar(stBar);
      3. stBar->addWidget(label);放左侧信息
      4. stBar->addPermanentWidget(label2); 放右侧信息
   4. 铆接部件 浮动窗口 可以多个
      1. QDockWidget
      2. addDockWidget( 默认停靠区域，浮动窗口指针)
      3. 设置后期停靠区域
   5. 设置核心部件 只能一个
      1. setCentralWidget(edit);
2. 资源文件
   1. 将图片文件 拷贝到项目位置下
   2. 右键项目->添加新文件 –> Qt - > Qt recourse File - >给资源文件起名
   3. res 生成 res.qrc
   4. open in editor 编辑资源
   5. 添加前缀 添加文件
   6. 使用 “ : + 前缀名 + 文件名 ”
3. 对话框
   1. 分类 ：
      1. 模态对话框 不可以对其他窗口进行操作 阻塞
         1. QDialog dlg(this)
         2. dlg.exec();
      2. 非模态对话框 可以对其他窗口进行操作
         1. 防止一闪而过 创建到堆区
         2. QDialog \* dlg = new QDialog(this)
         3. dlg->show();
         4. dlg2->setAttribute(Qt::WA\_DeleteOnClose); //55号 属性
   2. 标准对话框 -- 消息对话框
      1. QMessageBox 静态成员函数 创建对话框
      2. 错误、信息、提问、警告
      3. 参数1 父亲 参数2 标题 参数3 显示内容 参数4 按键类型 参数5 默认关联回车按键
      4. 返回值 也是StandardButton类型，利用返回值判断用户的输入
   3. 其他标准对话框
      1. 颜色对话框 QColorDialog：：getColor
      2. 文件对话框 QFileDialog：：getOpenFileName(父亲，标题，默认路径，过滤文件)
      3. 字体对话框 QFontDialog：：getFont
4. 界面布局
   1. 实现登陆窗口
   2. 利用布局方式 给窗口进行美化
   3. 选取 widget 进行布局 ，水平布局、垂直布局、栅格布局
   4. 给用户名、密码、登陆、退出按钮进行布局
   5. 默认窗口和控件之间 有9间隙，可以调整 layoutLeftMargin
   6. 利用弹簧进行布局
5. 控件
   1. 按钮组
      1. QPushButton 常用按钮
      2. QToolButton 工具按钮 用于显示图片，如图想显示文字，修改风格：toolButtonStyle ， 凸起风格autoRaise
      3. radioButton 单选按钮，设置默认 ui->rBtnMan->setChecked(true);
      4. checkbox多选按钮，监听状态，2 选中 1 半选 0 未选中
   2. QListWidget 列表容器
      1. QListWidgetItem \* item 一行内容
      2. ui->listWidget ->addItem ( item )
      3. 设置居中方式item->setTextAlignment(Qt::AlignHCenter);
      4. 可以利用addItems一次性添加整个诗内容
   3. QTreeWidget 树控件
      1. 设置头
         1. ui->treeWidget->setHeaderLabels(QStringList()<< "英雄"<< "英雄介绍");
      2. 创建根节点
         1. QTreeWidgetItem \* liItem = new QTreeWidgetItem(QStringList()<< "力量");
      3. 添加根节点 到 树控件上
         1. ui->treeWidget->addTopLevelItem(liItem);
      4. 添加子节点
         1. liItem->addChild(l1);
   4. QTableWidget 表格控件
      1. 设置列数
         1. ui->tableWidget->setColumnCount(3);
      2. 设置水平表头
         1. ui->tableWidget->setHorizontalHeaderLabels(QStringList()<<"姓名"<< "性别"<< "年龄");
      3. 设置行数
         1. ui->tableWidget->setRowCount(5);
      4. 设置正文
         1. ui->tableWidget->setItem(0,0, new QTableWidgetItem("亚瑟"));
   5. 其他控件介绍
      1. stackedWidget 栈控件
         1. ui->stackedWidget->setCurrentIndex(1);
      2. 下拉框
         1. ui->comboBox->addItem("奔驰");
      3. QLabel 显示图片
         1. ui->lbl\_Image->setPixmap(QPixmap(":/Image/butterfly.png"))
      4. QLabel显示动图 gif图片
         1. ui->lbl\_movie->setMovie(movie);
         2. movie->start();