**Material Style**

**导入库：import QtQuick.Controls.Material 2.12**

使用方法：

1.在qml中导入第一的模块

2.需要在main.cpp中设置

<1>在.pro文件在添加quickcontrols2说明

|  |
| --- |
| QT += quick quickcontrols2 |

<2>在main.cpp中包含头文件：

|  |
| --- |
| #include<QQuickStyle> |

<3> 在main.cpp中启用Material style

|  |
| --- |
| QQuickStyle::setStyle("Material"); |

简单使用：ApplicationWindow实例

|  |  |  |
| --- | --- | --- |
| //.pro文件  QT += quick quickcontrols2  CONFIG +=c++17  # You can make your code fail to compile if it uses deprecated APIs.  # In order to do so, uncomment the following line.  #DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0  SOURCES += \  main.cpp  RESOURCES += qml.qrc  # Additional import path used to resolve QML modules in Qt Creator's code model  QML\_IMPORT\_PATH =  # Additional import path used to resolve QML modules just for Qt Quick Designer  QML\_DESIGNER\_IMPORT\_PATH =  # Default rules for deployment.  qnx: target.path = /tmp/$${TARGET}/bin  else: unix:!android: target.path = /opt/$${TARGET}/bin  !isEmpty(target.path): INSTALLS += target | //main.cpp  #include <QGuiApplication>  #include <QQmlApplicationEngine>  #include<QQuickStyle>  int **main**(int **argc**, char \***argv**[])  {  QQuickStyle::setStyle("Material");  #if QT\_VERSION < QT\_VERSION\_CHECK(6, 0, 0)  QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);  #endif  QGuiApplication **app**(*argc*, *argv*);  QQmlApplicationEngine **engine**;  const QUrl **url**(QStringLiteral("qrc:/main.qml"));  QObject::connect(&engine, &QQmlApplicationEngine::objectCreated,  &app, [url](QObject \***obj**, const QUrl &**objUrl**) {  if (!obj && url == objUrl)  QCoreApplication::exit(-1);  }, Qt::QueuedConnection);  engine.load(url);  return app.exec();  } | //main.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQuick.Controls.Material 2.12  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello Material Style")  ApplicationWindow {  visible: true  Material.theme: Material.Dark  Material.accent: Material.Purple  Column {  anchors.centerIn: *parent*  RadioButton { text: *qsTr*("Small") }  RadioButton { text: *qsTr*("Medium"); checked: true }  RadioButton { text: *qsTr*("Large") }  }  }  } |

效果：

|  |
| --- |
|  |

按钮的简单使用

|  |  |  |
| --- | --- | --- |
| //.pro文件  QT += quick quickcontrols2  CONFIG +=c++17  # You can make your code fail to compile if it uses deprecated APIs.  # In order to do so, uncomment the following line.  #DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0  SOURCES += \  main.cpp  RESOURCES += qml.qrc  # Additional import path used to resolve QML modules in Qt Creator's code model  QML\_IMPORT\_PATH =  # Additional import path used to resolve QML modules just for Qt Quick Designer  QML\_DESIGNER\_IMPORT\_PATH =  # Default rules for deployment.  qnx: target.path = /tmp/$${TARGET}/bin  else: unix:!android: target.path = /opt/$${TARGET}/bin  !isEmpty(target.path): INSTALLS += target | //main.cpp  #include <QGuiApplication>  #include <QQmlApplicationEngine>  #include<QQuickStyle>  int **main**(int **argc**, char \***argv**[])  {  QQuickStyle::setStyle("Material");  #if QT\_VERSION < QT\_VERSION\_CHECK(6, 0, 0)  QCoreApplication::setAttribute(Qt::AA\_EnableHighDpiScaling);  #endif  QGuiApplication **app**(*argc*, *argv*);  QQmlApplicationEngine **engine**;  const QUrl **url**(QStringLiteral("qrc:/main.qml"));  QObject::connect(&engine, &QQmlApplicationEngine::objectCreated,  &app, [url](QObject \***obj**, const QUrl &**objUrl**) {  if (!obj && url == objUrl)  QCoreApplication::exit(-1);  }, Qt::QueuedConnection);  engine.load(url);  return app.exec();  } | //main.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQuick.Controls.Material 2.12  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello Material Style")// This is available in all editors.  Button{  text: "Button1"  highlighted: true  Material.accent: Material.Orange    }  } |

效果：

|  |
| --- |
|  |

适当修改qml文件

|  |  |
| --- | --- |
| //main.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQuick.Controls.Material 2.12  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello Material Style")// This is available in all editors.  Button{  text: "Button1"  highlighted: true  flat:true  Material.accent: Material.Orange  Material.background: Material.Teal  }  } | 效果： |

可以看到上例中的样式将文本全部转为大写，如果你不想全部大写，可以修改状态属性;

|  |  |
| --- | --- |
| //main.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  import QtQuick.Controls.Material 2.12  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello Material Style")// This is available in all editors.  Button{  text: "Button1"  font.capitalization: Font.MixedCase  highlighted: true  flat:true  Material.accent: Material.Orange  Material.background: Material.Teal  }  } | 效果： |

如果有兴趣，可以研究一下它的源码:

路径是：

|  |
| --- |
| C:\Qt\5.15.2\msvc2019\_64\qml\QtQuick\Controls.2\Material\ |

可以看看这里的文件，可以将源码中的设置拷贝到自己的项目中修改或者直接运行

/\*

阅读源码的难度

1.语法不认识

2.没有可以参照的文档

3.没有实际运行的效果测试

4.代码实际过于臃肿

5.抽象的太严重了

\*/