## 1.可以在qml文件中定义信号，就像在qt中一样。也是使用signal 信号名称(参数的方式)

实例1

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello World")  signal testSig(string s,int value)//定义信号  function *onTestSig*(s,value) //定义槽函数  {  *console*.log(*s* + " "+*value*)  }  Button{  width: 100  height: 30  background: Rectangle{  anchors.fill: *parent*  border.color:"yellow"  Text {  id: *txt*  text: *qsTr*("test")  anchors.centerIn: *parent*  font.pointSize: 16  }  }  onClicked: {  *testSig*("i am a custom signal",100) //可以通过点击按钮方式自定义信号，在qml中发送信号  }  }  Component.onCompleted: {  *testSig*.connect(*onTestSig*) //绑定槽函数  }  } |  |

### 以上的方法不好，还有更好的，使用qml的Connections控件

|  |  |
| --- | --- |
| import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  Window {  width: 640  height: 480  visible: true  title: *qsTr*("Hello World")  signal testSig(string s,int value)//定义信号  // function onTestSig(s,value) //定义槽函数，方法一需要  // {  // console.log(s + " "+value)  // }  Button{  y:*parent*.height-100  width: 100  height: 30  background: Rectangle{  anchors.fill: *parent*  border.color:"yellow"  Text {  id: *txt*  text: *qsTr*("test")  anchors.centerIn: *parent*  font.pointSize: 16  }  }  onClicked: {  *testSig*("i am a custom signal",100) //可以通过点击按钮方式自定义信号，在qml在发送信号  }  }  // Component.onCompleted: { 方法一  // testSig.connect(onTestSig) //绑定槽函数  // }  //方法二,使用Connections元素  Image {  id: *img*  width:200  height: 300  source: "images/gzmsn.gif"  }  Connections{  target: *window*  function *onTestSig*(s,v){  //console.log(s + ":" +v);  // img.width += 5;  // img.height +=5;  *img*.x +=50;  if(*img*.x > *window*.width){  *img*.x = 0;  }  }  }  } |  |

## 2.自定义控件的信号和槽

|  |  |  |
| --- | --- | --- |
| //main.qml  ort QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  Window {  id:*window*  width: 640  height: 480  visible: true  title: *qsTr*("Hello World")  Component{  id:*com*  Button{  width: 100  height:40  signal btnSig(int val)  onClicked: {  // console.log("i am clicked...")  *btnSig*(100)  }  }  }  MyComponent{  com1: *com*  com2: *com*  }  } | //MyComponent.qml  import QtQuick 2.15  import QtQuick.Window 2.15  import QtQuick.Controls 2.15  Rectangle{  width: 400  height: 300  property Component com1  property Component com2  border.color: "purple"  Loader{  id:*loader1*  sourceComponent: *com1*  anchors.bottom: *parent*.bottom  anchors.bottomMargin: 20  anchors.right: *parent*.right  anchors.rightMargin: 20  Connections{  target: *loader1*.item  ignoreUnknownSignals: true  function onBtnSig(val){  *console*.log("from com1: "+*val*)  }  }  }  Loader{  id:*loader2*  sourceComponent: *com2*  anchors.bottom: *parent*.bottom  anchors.bottomMargin: 20  anchors.right: *parent*.right  anchors.rightMargin: 140  Connections{  target: *loader2*.item  ignoreUnknownSignals: true  function onBtnSig(val){  *console*.log("from com2: "+*val*)  }  }  }  } |  |