

TextInput :

```
Rectangle {
  id:rect
  width:100
  height:20
  anchors.top: label.top
  anchors.left : label.right
  // anchors.top: label.bottom
  // anchors.left : label.left
  border.width : 2
  border.color : "black"
  TextInput {
    id: input1
    anchors.fill: rect
  }
}
```

Button:

```
Rectangle {
  id:button
  width:100
  height:40
  anchors.top: rect.bottom
  anchors.left : rect.left
  border.width : 2
  border.color : "black"
  color:"blue"
  Text {
    anchors.fill:button
    text : "click me"
  }
  MouseArea {
    anchors.fill:button
    onClicked: {
      console.log("button clicked:"
                  + input1.text)
    }
  }
}
```

Light.qml:-

```
import QtQuick 2.0

Rectangle {
    property string bgcolor: "black"
    property int dia: 200
    id : light
    width : dia
    height : dia
    radius : dia/2
    color : bgcolor
    border.width: 2
    border.color: "black"
}
```

```
Column {
    spacing:20
    Light {
        id : rlight
        bgcolor: "red"
        dia : 200
    }
    Light {
        id : ylight
        bgcolor: "yellow"
        dia : 200
    }
    Light {
        id : glight
        bgcolor: "green"
        dia : 200
    }
}
```

Timer code:-

```
Timer {  
    id : timer  
    interval: 2000  
    running: true  
    repeat: true  
    onTriggered: {  
        if(k%3==0) {  
            rlight.bgcolor = "black"  
            ylight.bgcolor = "yellow"  
            glight.bgcolor = "black"  
        }  
        else if(k%3==1) {  
            rlight.bgcolor = "black"  
            ylight.bgcolor = "black"  
            glight.bgcolor = "green"  
        }  
        else {  
            rlight.bgcolor = "red"  
            ylight.bgcolor = "black"  
            glight.bgcolor = "black"  
        }  
        k++;  
    }  
}
```

Quick Controls:-

```
import QtQuick.Controls 2.5
```

```
Column {  
    spacing: 20  
    Slider {  
        id : slider  
        from: 0  
        to : 100  
        value : 32  
        onValueChanged: {  
            pbar.value = slider.value  
        }  
    }  
    ProgressBar {  
        id : pbar  
        from : 0  
        to : 100  
        value : 64  
    }  
}
```

Basic Controls

Layouts

<https://doc.qt.io/qt-5/qtquicklayouts-index.html>

Styles

Extras

Further:-

- * Animations

- * Cpp & QML Integrations

Animations -- change in property

```
SequentialAnimation {  
  running: true  
  loops : 5  //Animation.Infinite  
  PropertyAnimation {  
    target : ball  
    duration : 1000  
    property : "x"  
    to : 700  
  }  
  PropertyAnimation {  
    target : ball  
    duration : 1000  
    property : "y"  
    to : 500  
  }  
}
```

```
ParallelAnimation {  
  running: true  
  loops : 5  //Animation.Infinite  
  PropertyAnimation {  
    target : ball  
    duration : 1000  
    property : "x"  
    to : 700  
  }  
  PropertyAnimation {  
    target : ball  
    duration : 1000  
    property : "y"  
    to : 500  
  }  
}
```

SequentialAnimation {
 ParallelAnimation {
 // x: 0 to 700, y : 0 to 500
 }
 ParallelAnimation {
 // x : 700 to 0, y : 500 to 0
 }
}

High level:-

- * Sequential
- * Parallel

Low level:-

- * Number Animation
- * Color Animation
- * Rotation Animation
- * Property Animation

```
SequentialAnimation {  
  running: true  
  loops: Animation.Infinite  
  RotationAnimation {  
    duration: 1000;  
    direction: RotationAnimation.Clockwise  
    target: box  
    to: 180  
  }  
}
```