1. Araştırma

Qml Templates:

1. List

2. Swipe

3. Stack

4. Blank

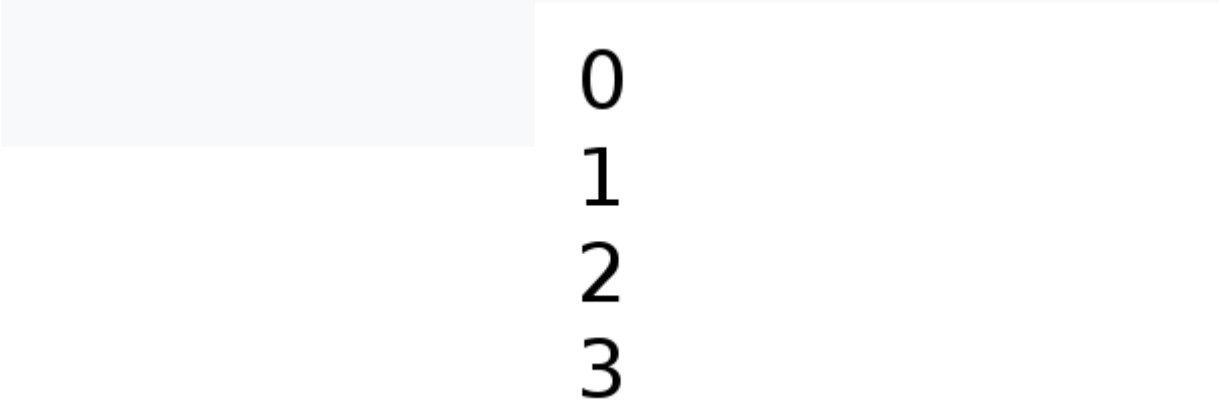
QML'de Dinamik Nesne ve Liste Yönetimi

QML, QML nesnelerini dinamik olarak oluşturmak ve yönetmek için bir dizi yol sağlar. Loader, Repeater, **ListView**, **GridView** ve **PathView** öğelerinin tümü dinamik nesne yönetimini destekler. Nesneler ayrıca C ++ 'dan oluşturulabilir ve yönetilebilir ve bu, hibrit QML / C ++ uygulamaları için tercih edilen yöntemdir (bkz. C ++ Uygulamalarında QML Bağlamalarını Kullanma).

QML, nesnelerin JavaScript kodu içinden dinamik olarak oluşturulmasını da destekler. Bu, mevcut QML öğeleri uygulamanızın gereksinimlerine uymuyorsa ve ilgili C ++ bileşeni yoksa yararlıdır.

ListView:

ListView, öğelerin yatay veya dikey olarak yerleştirildiği klasik bir öğe listesi gösterir. Figür 1, bir sayı dizisini görüntüleyen minimum bir **ListView** gösterir (model olarak bir tamsayı kullanarak). Modeldeki her veri parçası için bir öğe tanımlamak için basit bir temsilci kullanılır.



Figür 1. ListView örnek.

Figür 1 Kod:

```
import QtQuick 1.0

ListView {
    width: 50; height: 200
    model: 4
    delegate: Text {
        text: index;
        font.pixelSize: 40
    }
}
```

GridView: GridView, öğeleri bir dosya yöneticisinin simge görünümü gibi bir gridde görüntüler.

GridView Önek Kod:

```
GridView {
        id: grid
        anchors.fill: parent
        cellWidth: 80; cellHeight: 80

        model: ContactModel {}
        delegate: contactDelegate
        highlight: Rectangle { color: "lightsteelblue"; radius: 5 }
        focus: true
    }
```

PathView: PathView, seçimin aynı yerde kaldığı ve öğelerin onun etrafında hareket ettiği bir path üzerindeki öğeleri görüntüler.

PathView Önek Kod:

```
PathView {
          anchors.fill: parent
          model: ContactModel {}
          delegate: delegate
          path: Path {
              startX: 120; startY: 100
              PathQuad { x: 120; y: 25; controlX: 260; controlY: 75 }
              PathQuad { x: 120; y: 100; controlX: -20; controlY: 75 }
      }
```

2. Çalışma ve Kazanımlar (Component: 5.15.1 MinGw)

Kod:

```
import QtQuick 2.12
import QtQuick.Window 2.12

//ID - unique!!
//Root Object - only one!!

Window {
    width: 640
    height: 480
    visible: true
    title: qsTr("Hello World")
    id:root



    Image {
        id: myImage

        source:"http://upload.wikimedia.org/wikipedia/commons/0/0b/Qt_logo_
        2016.svg" //As a default, it has to be files.
        //It works with the links cash!!
        width: 150
        height: 100
        anchors.centerIn: parent //placement

        Rectangle{
            color:"red"
            width: parent.width
            height: parent.height
            opacity: 0.5

    }



    }

}
```

Output: