Udemy - Qt Quick and QML For Beginners the Fundamentals

# Introduction

**Qt Quick**- Qt User Interface Creational Kit.

**QML**- Qt Markup Language

Graphical user interface, website

Description automatically generated

A picture containing diagram

Description automatically generated

# Install Qt and Qt Creator

Go to <http://download.qt.io/official_releases/qt/> and download appropriate file for your platform.

For Linux:

Install following packages before run app.

$ sudo apt-get install mesa-common-dev

$ sudo apt-get install libglu1-mesa-dev

Qt Widget Demo App

File 🡪 New File or Project 🡪 Application (Qt) 🡪 Qt Widgets Application 🡪 Choose 🡪 Name of Project 🡪 Build System = qmake 🡪 Next 🡪 Change base Class to “QWidget” 🡪 Next … Next 🡪 Finish

Qt Global Object:

* The Qt Object is a global object with utility functions, properties and enums.
* It is not Instantiable.

**Every Property in QML have auto generated handler with name as prefix “on” +capital 1st letter of prop +suffix “Changed”**

e.g.

property string myProp: "Vijay"

onMyPropChanged: {

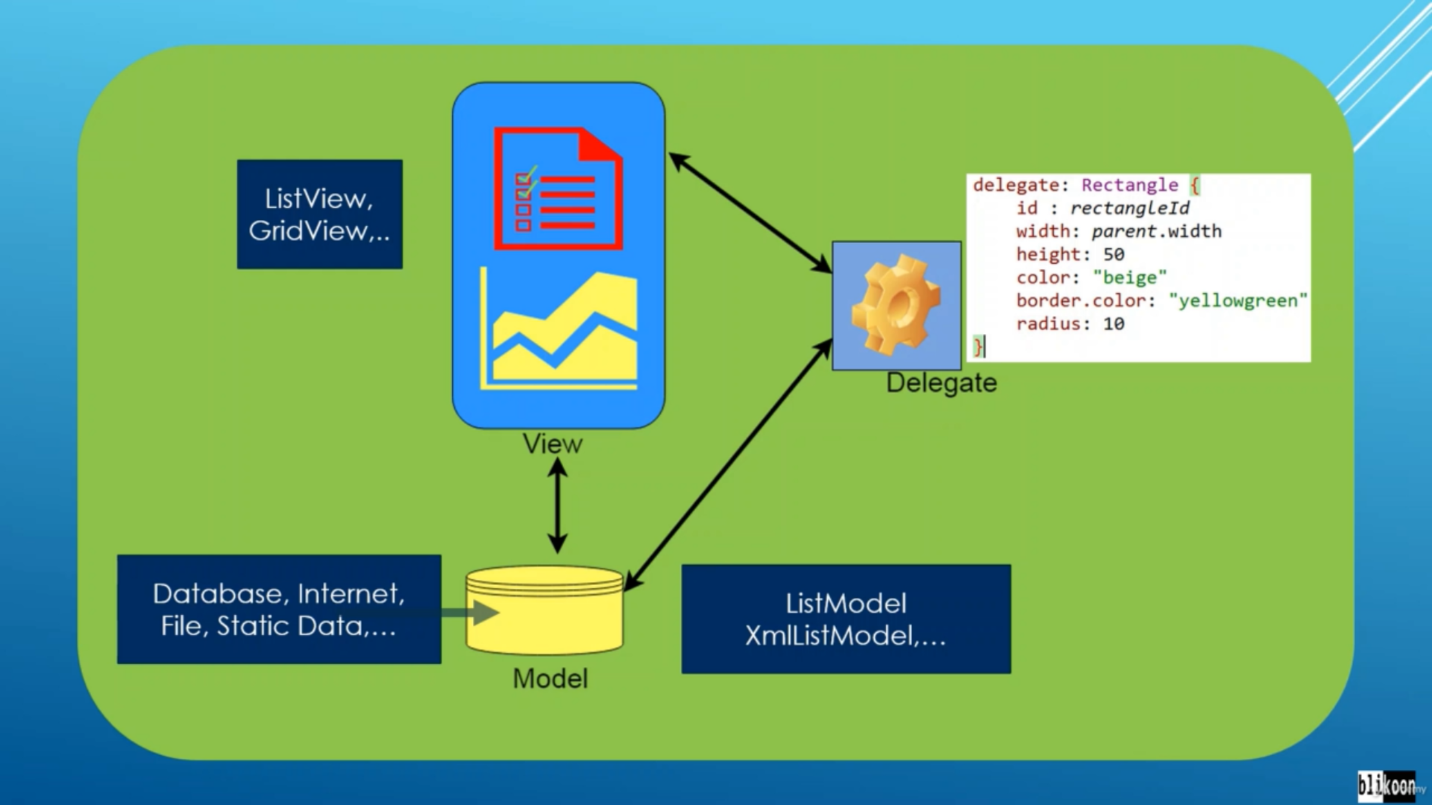
*console*.log("myProp changed to "+*myProp*)

}

# Signal and Slot

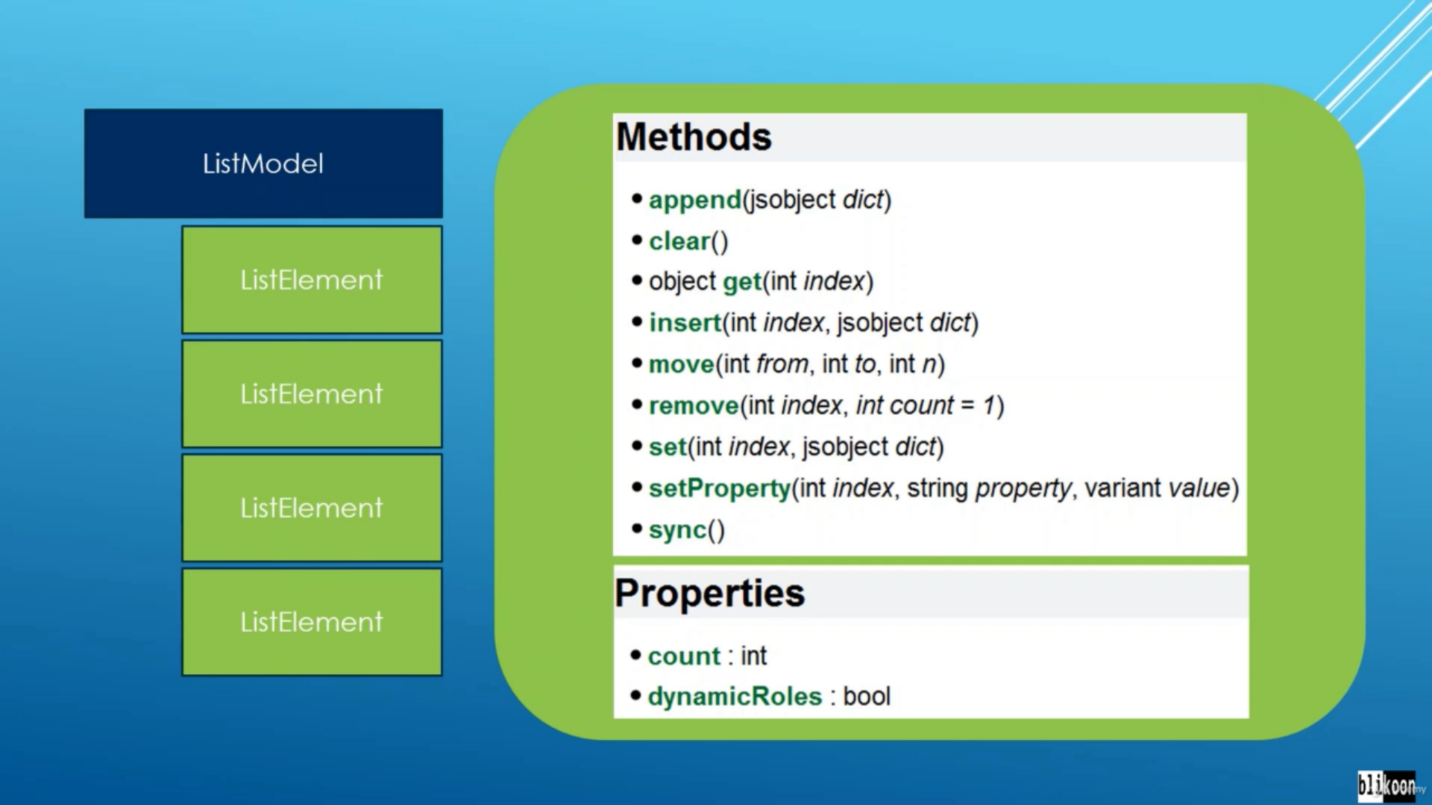
* **Declaration**: *signal signalName(args)*
* When you declare a signal, a special automatic handler is generated. You can call the handler using the format ***on<SignalName>***
* **Slot**: A method that is called as a result of signal being fired
* Connect a slot: <rootId>.<SignalName>.connect(<rootId>.<function\_name\_to\_connect>)

# Model, View and Delegate Architecture:



# Model Types

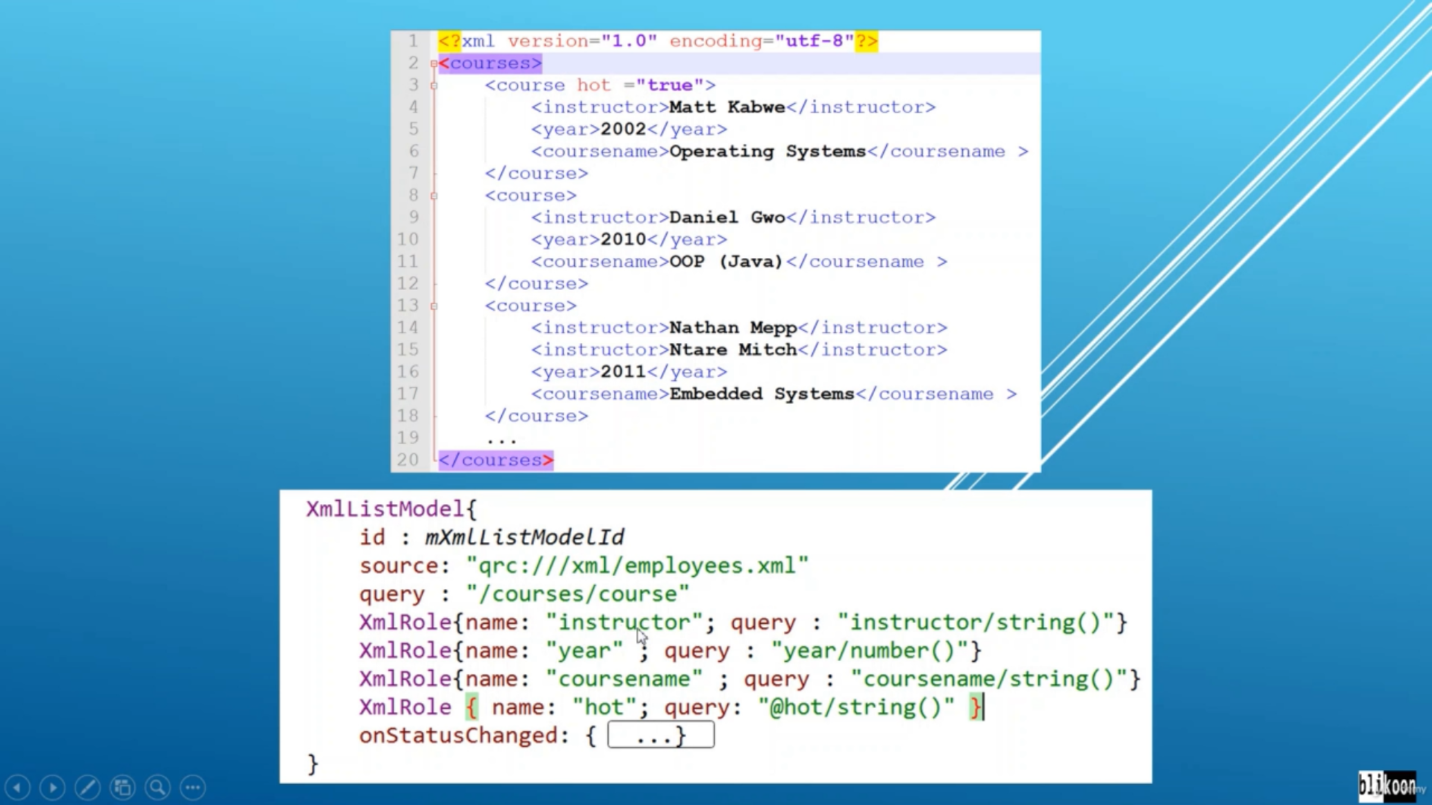
## List Model

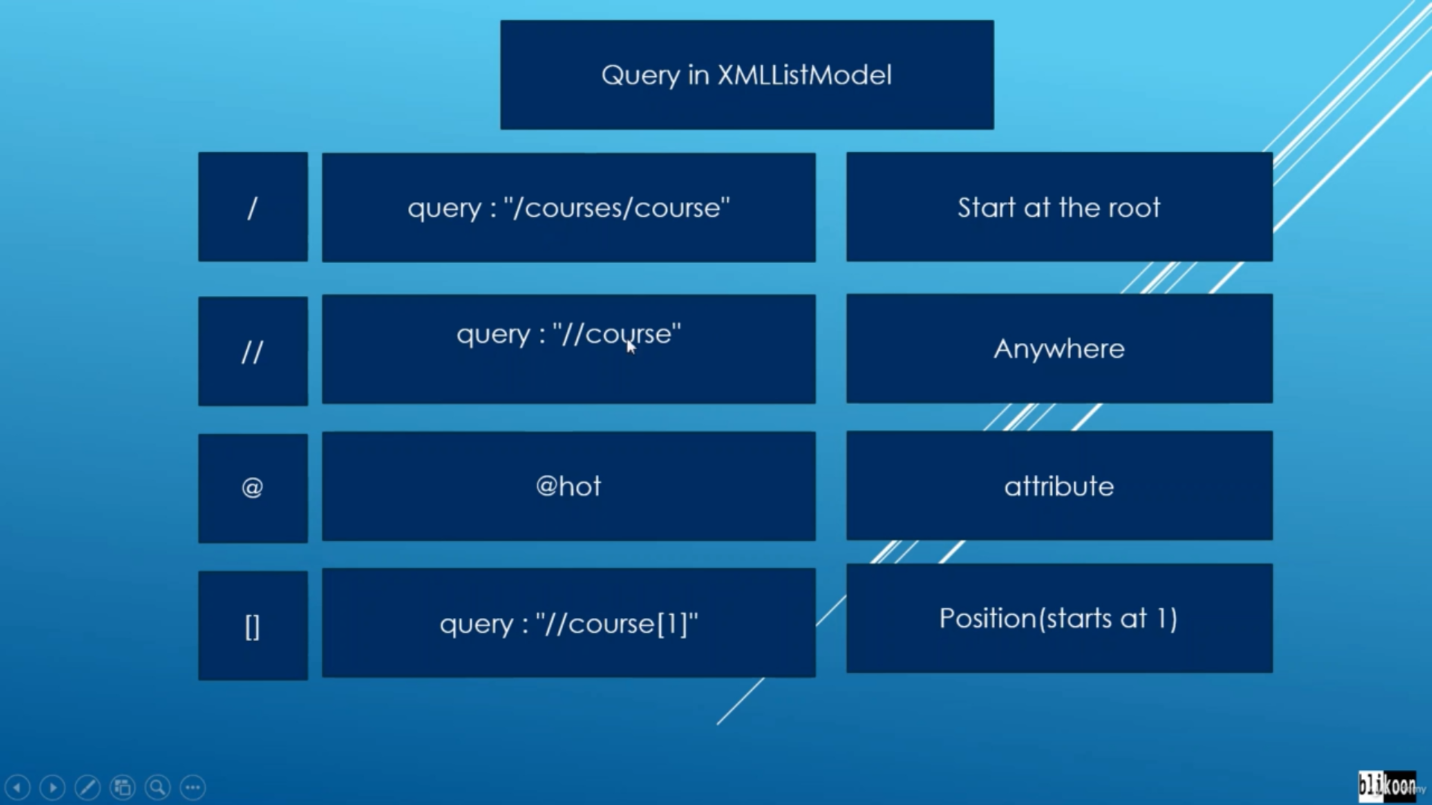


## Xml List Model

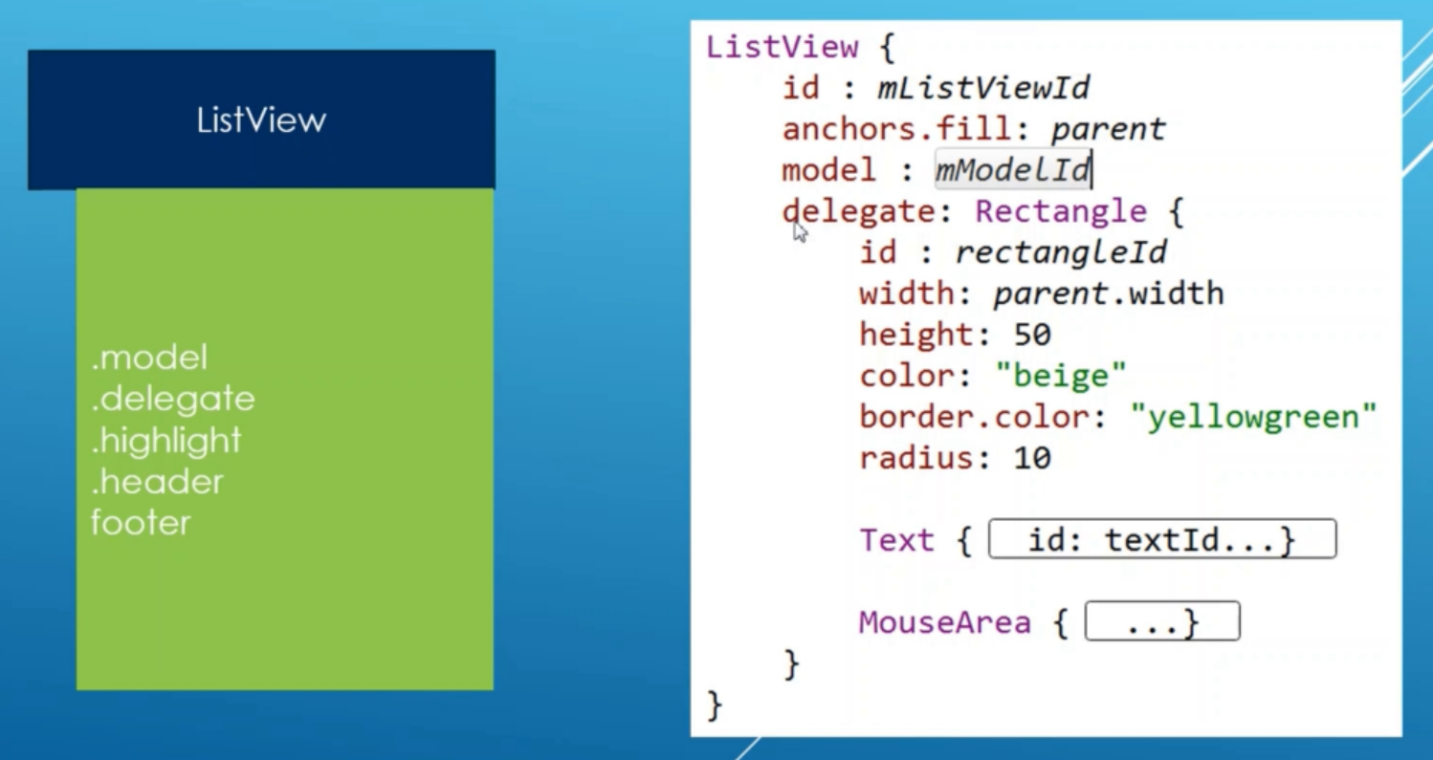


### Read data from Xml file to Model:





# List View:



# Networking in Qml

