Notes to self

```
. Exploring the exchange of sequence data
   . There are many types on the C++ side
       . QVector
       . QList
       .....
    . Most of them map to simple JS arrays on the QML side.

    . We  :
       . Send sequence data to C++ :
           Button{
               id : mButton1
               text : "Send to C++"
               onClicked: {
                   var arr = ['Apple', 'Banana','Avocado','Pear','Orange'];
                   CppClass.qmlArrayToCpp(arr)
               }
           }

       . Receive sequence data from C++ :

           Button{
               id : mButton2
               anchors.top: mButton1.bottom
               text : "Read from C++"
               onClicked: {
                   var arr = CppClass.retrieveStrings();
                   print("The length is : "+ arr.length)

                   arr.forEach(function(element){
                       console.log(element)
                   })
               }
           }
```

Sequence Types from C++ Turn to JS Arrays

The C++ Side

```
CppClass::CppClass(QObject *parent) : QObject(parent)
{
    mVector.append("One");
    mVector.append("Two");//...
}

void CppClass::qmlArrayToCpp(QVector<QString> vector)
{
    foreach (QString string, vector) {
        qDebug() << string;
    }

}

QVector<QString> CppClass::retrieveStrings()
{
    return mVector;
}
```

The QML Side

```
Button{
    text : "Send to C++"
    onClicked: {
        var arr = ['Apple', 'Banana','Avocado','Pear','Orange'];
        CppClass.qmlArrayToCpp(arr)
    }
}

Button{
    text : "Read from C++"
    onClicked: {
        var arr = CppClass.retrieveStrings();
        arr.forEach(function(element){
            console.log(element)
        })
    }
}
```