Notes to self

```
. Exploring property value sources :
    . The C++ behind the syntax like :
        NumberAnimation on x { to: 50; duration: 1000 }

    . NumberAnimation is a property value source because the values
      that go into x come from it.

    . We will be building a random number class, that will act as a property
      value source for other types.

    . The flow :
          . You create a custom class
          . The class will need to inherit QQmlPropertyValueSource
                class RandomNumber : public QObject,public QQmlPropertyValueSource
          . The class needs to provide the macro
                Q_INTERFACES(QQmlPropertyValueSource)
          . The class needs to provide a setTarget method :
                virtual void setTarget(const QQmlProperty & prop);

          . The class needs to provide an updateProperty slot :
              void updateProperty();

          . QQmlProperty has a bunch of methods you can use to forward
              and pull data to/from the property wrapped around.

          . And what we wrap around is what our Random number bounds to
              on the QML side, for example, we may be bounding to the
              radius or width of a rectangle.

          . In updateProperty, we write to the property and what we write
              is our generated random number.

          . With the plumbing in place, the rest is :

              . exposing to qml :
                  .     qmlRegisterType<RandomNumber>("RandomUtil",1,0,"RandomNumber");

              . using the types in QML :
                      Rectangle{
                          id : mRect
                          RandomNumber on width {
                              maxValue: 700
                          }
                          height: 300
                          color: "dodgerblue"
                          RandomNumber on radius {
                              maxValue: 300
                          }
                      }

        . Use the explanations in the Qt 5 course and improvise.
```

Property Value Sources

- The C++ behind the syntax like

```
NumberAnimation on x { to: 50; duration: 1000 }
```

- NumberAnimation is a property value source because the values that go into x come from it

```
class RandomNumber : public QObject,public QQmlPropertyValueSource
{
    Q_OBJECT
    Q_PROPERTY(int  maxValue READ maxValue WRITE setMaxValue NOTIFY maxValueChanged)
    Q_INTERFACES(QQmlPropertyValueSource)
public:
    explicit RandomNumber(QObject *parent = nullptr);
    virtual void setTarget(const QQmlProperty & prop);
    int maxValue() const;
    void setMaxValue(int maxValue);
signals:
    void maxValueChanged(int maxValue);
private slots :
    void updateProperty();
private:
    QQmlProperty m_targetProperty;
    int m_maxValue;
    QTimer * m_timer;
};
```

Update the property

```
void RandomNumber::updateProperty()
{
    m_targetProperty.write(QRandomGenerator::global()->bounded(m_maxValue));
}
```

Register the types

```
qmlRegisterType<RandomNumber>("RandomUtil",1,0,"RandomNumber");
```

QML

```
Rectangle{
    id : mRect
    RandomNumber on width {
        maxValue: 700
    }
    height: 300
    color: "dodgerblue"
    RandomNumber on radius {
        maxValue: 300
    }
}
```