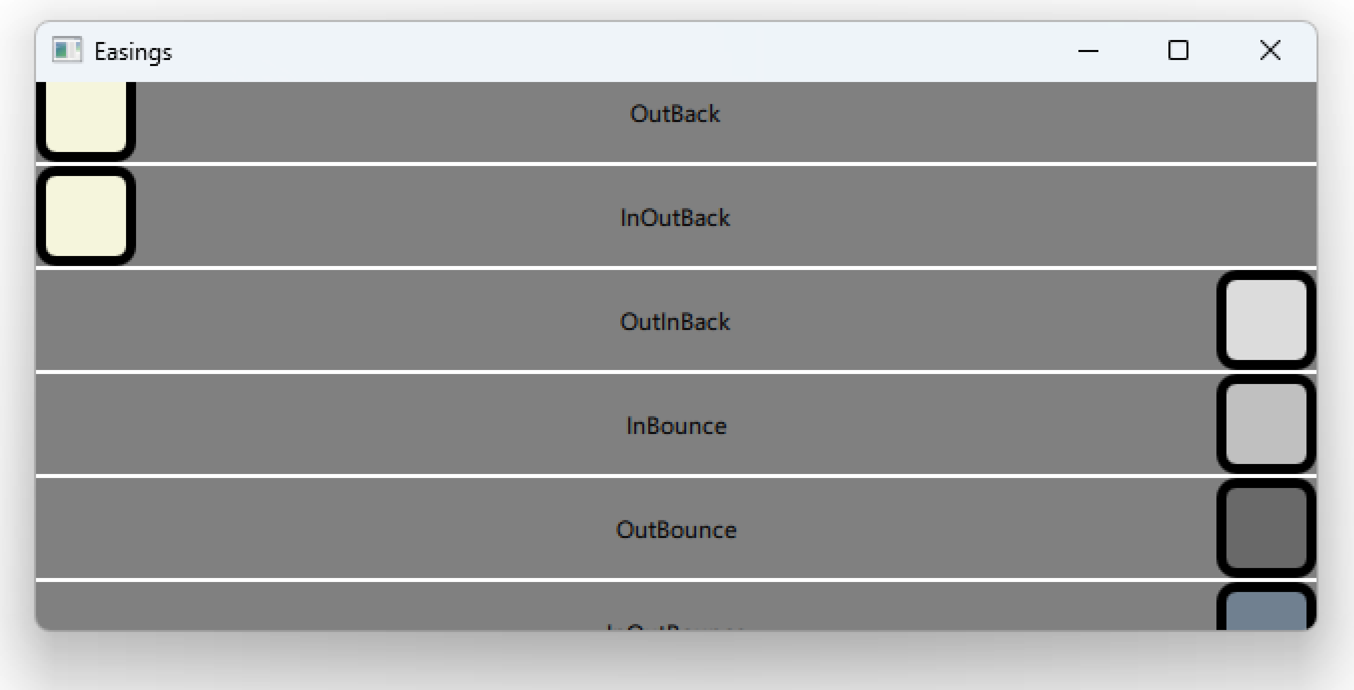
# Notes to self

```
. Exploring easing curves
. Easing curves control how the animation changes over time, for example, how
    fast we go from the beginning to the end.
. Improvise.
```

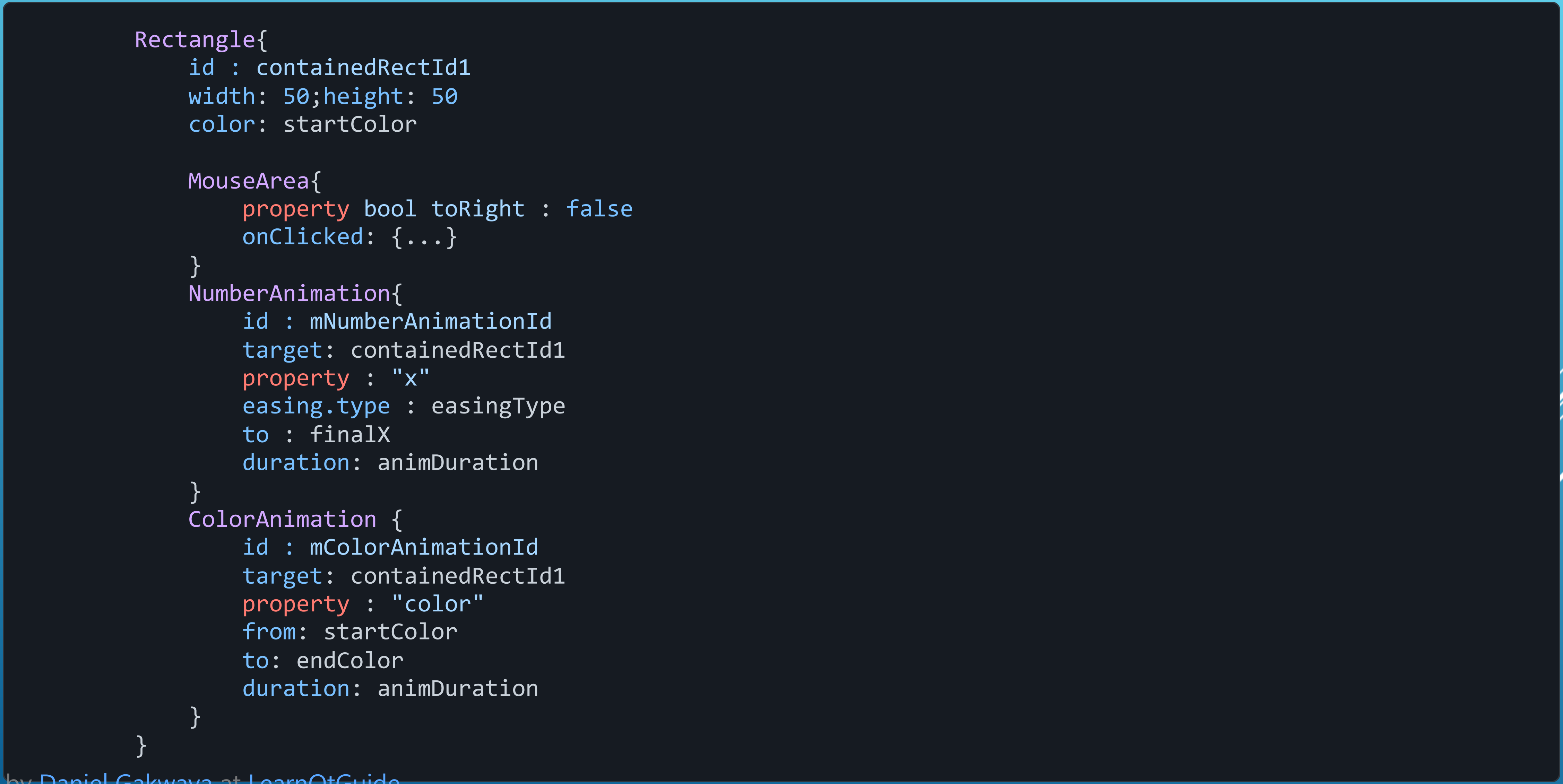
# Easing Curves



# Purpose

- Controlling how the animation goes from start to finish

# Animated Rect

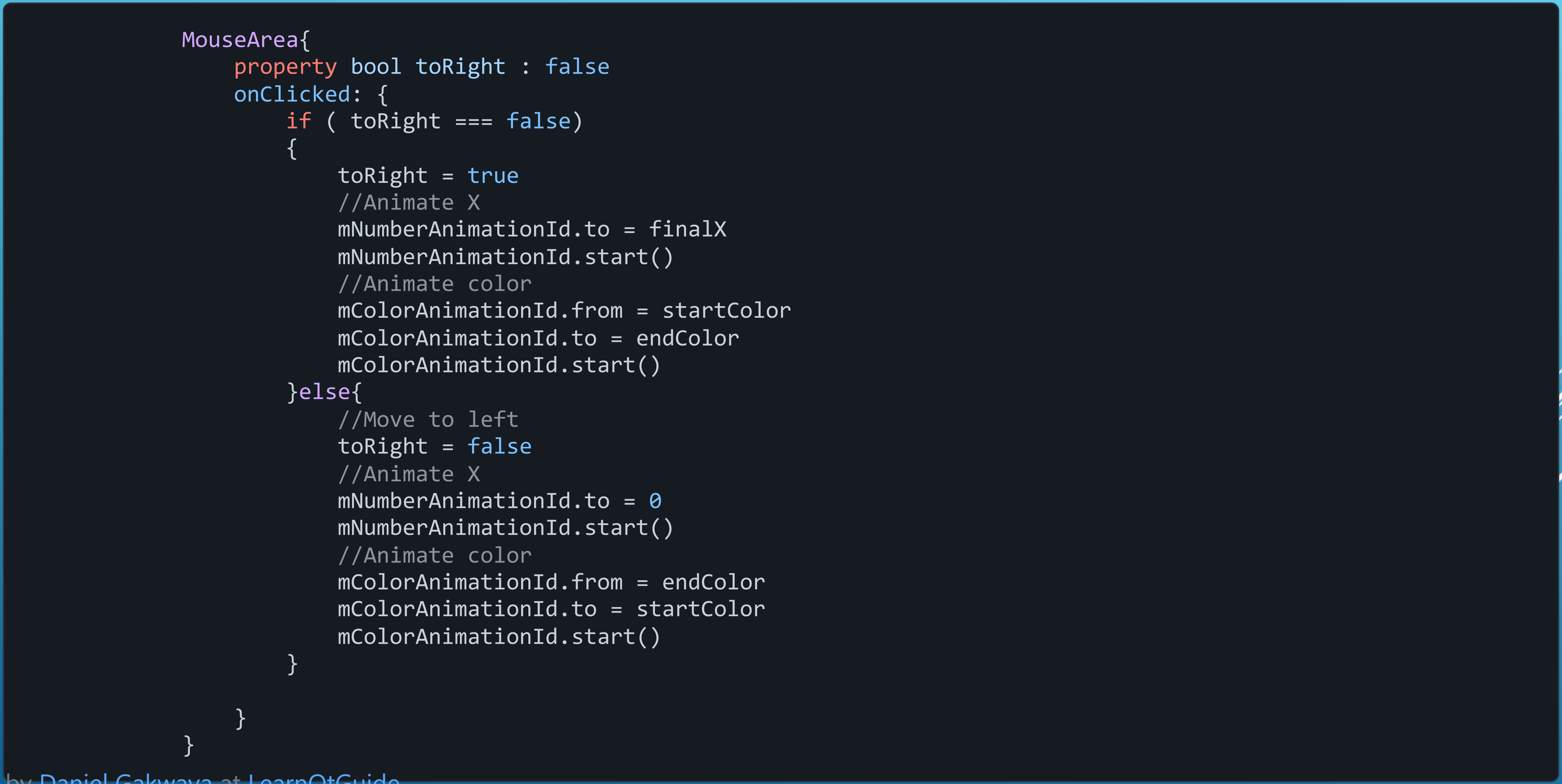


```
Rectangle{
    id : containedRectId1
    width: 50;height: 50
    color: startColor

    MouseArea{
        property bool toRight : false
        onClicked: {...}
    }
    NumberAnimation{
        id : mNumberAnimationId
        target: containedRectId1
        property : "x"
        easing.type : easingType
        to : finalX
        duration: animDuration
    }
    ColorAnimation {
        id : mColorAnimationId
        target: containedRectId1
        property : "color"
        from: startColor
        to: endColor
        duration: animDuration
    }
}
```

Slides by Daniel Gakwaya at LearnQtGuide

# Handling Clicks



```
MouseArea{
    property bool toRight : false
    onClicked: {
        if ( toRight === false)
        {
            toRight = true
            //Animate X
            mNumberAnimationId.to = finalX
            mNumberAnimationId.start()
            //Animate color
            mColorAnimationId.from = startColor
            mColorAnimationId.to = endColor
            mColorAnimationId.start()
        }else{
            //Move to left
            toRight = false
            //Animate X
            mNumberAnimationId.to = 0
            mNumberAnimationId.start()
            //Animate color
            mColorAnimationId.from = endColor
            mColorAnimationId.to = startColor
            mColorAnimationId.start()
        }

    }
}
```

# Exported Properties

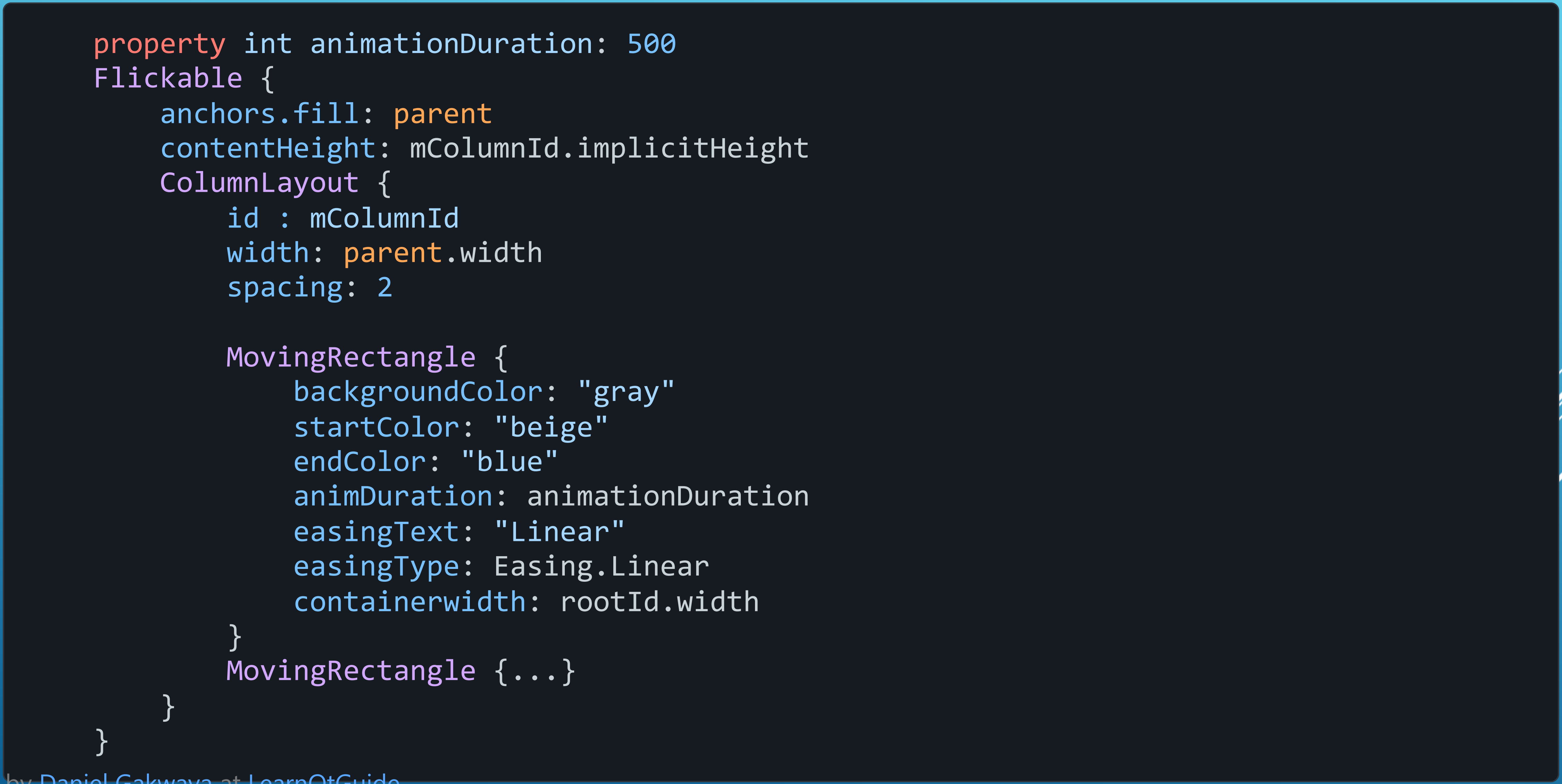
```
import QtQuick

Item {

    property var backgroundColor
    property var startColor
    property var endColor
    property string easingText
    property int animDuration
    property var easingType
    property int containerwidth
    width: smallContainerRectId1.width
    height: smallContainerRectId1.height
    property int finalX: smallContainerRectId1.width - containedRectId1.width

    Rectangle {
        id : smallContainerRectId1
        width: containerwidth
        height: 50
        color: backgroundColor
        Text {
            text: easingText
            anchors.centerIn: parent
        }
    }
}
```

# Create MovingRect Objects



```
property int animationDuration: 500
Flickable {
    anchors.fill: parent
    contentHeight: mColumnId.implicitHeight
    ColumnLayout {
        id : mColumnId
        width: parent.width
        spacing: 2

        MovingRectangle {
            backgroundColor: "gray"
            startColor: "beige"
            endColor: "blue"
            animDuration: animationDuration
            easingText: "Linear"
            easingType: Easing.Linear
            containerwidth: rootId.width
        }
        MovingRectangle {...}
    }
}
```

# CMake

```
find_package(Qt6 6.2 COMPONENTS Quick QuickControls2 REQUIRED)
...
target_link_libraries(app2-Button
    PRIVATE Qt6::Quick Qt6::QuickControls2)
```