

## Item QML Type:

Item türü, Qt Quick'taki tüm görsel öğeler için temel türdür. Qt Quick'taki tüm görsel öğeler, Öğ'e'den devralınır. Bir Item nesnesinin görsel bir görünümü olmasa da, x ve y konumu, genişlik ve yükseklik, sabitleme ve anahtar işleme desteği gibi görsel öğeler arasında ortak olan tüm nitelikleri tanımlar. Kordinat kullanımı, genişlik ve yükseklik kavramlarını örneklerle önceki raporlarda kullanılmıştır. Bu program dikdörtgenden bir daire elde etmek için yazılmıştır. QML de daire yoktur!!

## KOD:

```
import QtQuick 2.12
import QtQuick.Window 2.12

//Non-visual Item Class
//There is no cicle in QML, it is made by RECTANGLE in QML.(It is an item.)

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

    Rectangle{
        id:myItem
        anchors.centerIn: parent
        color: "red"
        width: 200
        height: 200
        visible: true

        border.color: "black"
        border.width: 5

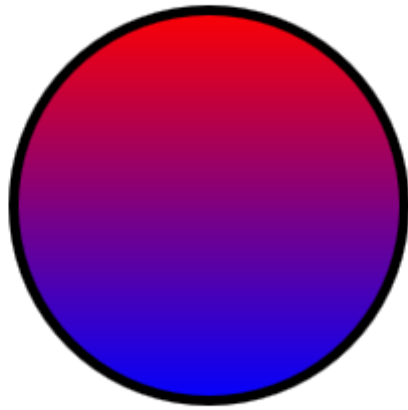
        radius: width //cicle

        gradient: Gradient{ //property
            GradientStop {position: 0.0; color: "red"}
            GradientStop {position: 1.0; color: "blue"}

        }

    }
}
```

**OUTPUT:**



## MouseArea QML Type:

MouseArea, genellikle bir öğe için fare kullanımı sağlamak, görünür bir öğe ile birlikte kullanılan görünmez bir öğedir. Etkili bir proxy olarak hareket ederek, fare işleme mantığı bir MouseArea öğesinin içinde yer alabilir.

### KOD:

```
import QtQuick 2.12
import QtQuick.Window 2.12

//MouseArea.
// TapHandler.

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

    Rectangle{
        id: myArea
        color: "blue"
        width: 200
        height: 200
        anchors.centerIn: parent

        MouseArea{
            id: myMouse
            anchors.fill: parent //to click the Rectangle
            acceptedButtons: Qt.LeftButton | Qt.RightButton // To click
right and left.
            hovered: true // to work with "onPositionChanged".
            onClicked: {
                console.log("Clicked: " + mouse.button)
                if(mouse.button === Qt.LeftButton) parent.color = "green"
                if(mouse.button === Qt.RightButton) parent.color = "red"
            }

            onDoubleClicked: console.log("Double Clicked: " + mouse.button)
            onPositionChanged: console.log("Position X: " + mouseX + "Y: "
+ mouseY) //Gives position of the mouse with x and y coordinates.
            onEntered: parent.color = "orange" // Mouse in.
            onExited: parent.color = "blue" // Mouse out.
        }
    }
}
```

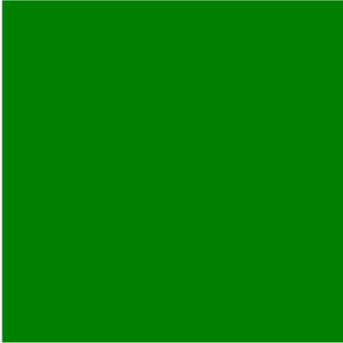
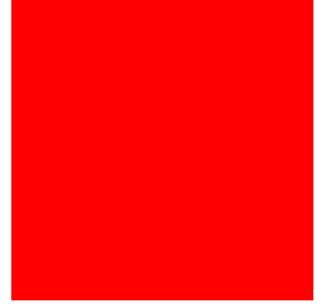
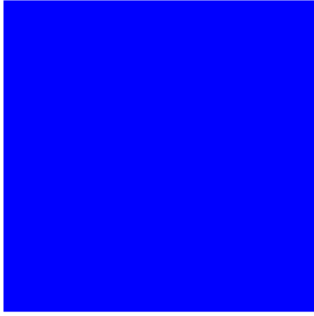
## OUTPUT:

Sol Tık → yeşil

Sağ Tık → kırmızı

Mouse içerde → turuncu

Mouse dışarıda → mavi



## Console Output:

qml: Position X: 80Y: 134

qml: Position X: 80Y: 133

qml: Position X: 81Y: 133

qml: Clicked: 2

qml: Clicked: 1

qml: Clicked: 1

qml: Double Clicked: 1

qml: Clicked: 2

qml: Clicked: 1

qml: Clicked: 1

qml: Double Clicked: 1

qml: Position X: 81Y: 132

qml: Position X: 81Y: 131

qml: Clicked: 2

qml: Double Clicked: 2

qml: Position X: 80Y: 131

qml: Clicked: 1

qml: Double Clicked: 1

qml: Clicked: 2

## Bazı MouseArea Sinyalleri:

1. [onCanceled\(\)](#)
2. [onClicked\(MouseEvent mouse\)](#)
3. [onDoubleClicked\(MouseEvent mouse\)](#)
4. [onEntered\(\)](#)
5. [onExited\(\)](#)