

Introducción a QML - III

Veamos ahora algo más de teoría, centrémonos en el UI:

Elementos QML

Un Elemento es la base de todos los componentes del GUI

Las propiedades básicas son:

- Coordenadas: x, y, z, **width**, **height**, **anchors**
- Transformaciones: **rotation**, **scale**, **translate**
- Jerarquía: **children**, **parent**
- Visibilidad: **visible**, **opacity**
- Estado: **state** y **transitions**

Un elemento QML por si solo no dibuja nada

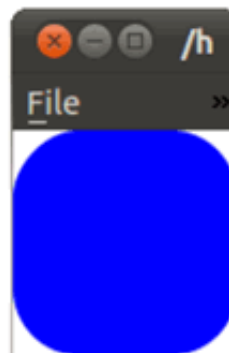
Elementos básicos QML

Como elementos básicos tenemos:

- Rectángulo (**Rectangle**)
- Imagen (**Image**): Las Imágenes se pueden descargar de la web
- Texto no editable (**Text**)
- Texto editable de una sola línea(**TextInput**)
- Texto editable multi-línea (**TextEdit**)
- Otros en desarrollo...

```
import Qt 4.7

Rectangle {
    width: 100
    height: 100
    color: "blue"
    radius: 30
}
```



```
Image {  
  width: 100  
  height: 100  
  source: "http://qt.n  
}
```



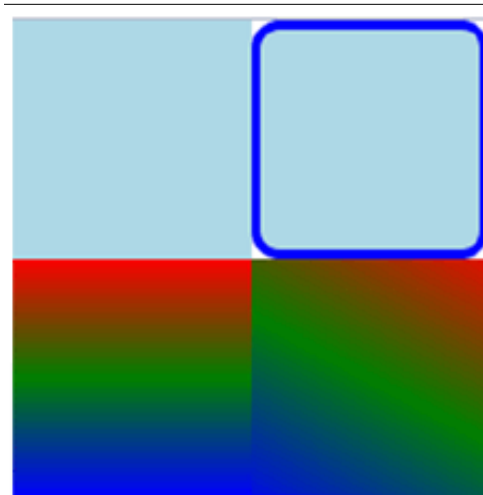
```
Rectangle {  
  width: 100  
  height: 100  
  
  TextEdit {  
    anchors.fill: parent  
    anchors.margins: 10  
    wrapMode: TextEdit.WordWrap  
  }  
}
```



Rectángulo (Rectangle)

Es un elemento esencial. Como propiedades básicas del mismo tenemos:

- **Color** (color): Puede ser una referencia mediante un nombre o un valor hexadecimal.
- **Gradiente** (gradient): El gradiente es siempre vertical y se puede rotar.
- **Borde** (border)
- **Radio** (radius)



Imágen (Image)

Es otro elemento esencial para representar imágenes en los estándares habituales (jpg, png, svg)...

- Podemos asociar desde ficheros o direcciones URL
- cachear las imágenes (**cache**)
- Carga las imágenes web de forma asíncrona (**asynchronous**)
- Podemos cambiar el tamaño de la imagen mediante las propiedades de sourceSize (Elemento QSize): (ancho) **width** y (alto) **height**.
- Establecer el modo de relleno (**fillMode**): Image.**Stretch**, Image.**PreserveAspectFit**, Image.**PreserveAspectCrop**, Image.**Tile**, Image.**TileVertically**, Image.**TileHorizontally**
- Hereda de **AnimatedImage**



Stretch (default)

```
Image {  
    width: 130; height: 100  
    smooth: true  
    source: "qtlogo.png"  
}
```



PreserveAspectRatio

```
Image {  
    width: 130; height: 100  
    fillMode: Image.PreserveAspectRatio  
    smooth: true  
    source: "qtlogo.png"  
}
```



PreserveAspectCrop

```
Image {  
    width: 130; height: 100  
    fillMode: Image.PreserveAspectCrop  
    smooth: true  
    source: "qtlogo.png"  
    clip: true  
}
```



Tile

```
Image {  
    width: 120; height: 120  
    fillMode: Image.Tile  
    source: "qtlogo.png"  
}
```



TileVertically

```
Image {  
    width: 120; height: 120  
    fillMode: Image.TileVertically  
    smooth: true  
    source: "qtlogo.png"  
}
```



TileHorizontally

```
Image {  
    width: 120; height: 120  
    fillMode: Image.TileHorizontally  
    smooth: true  
    source: "qtlogo.png"  
}
```

Text.AlignTop

Text.AlignVCenter

Text.AlignBottom

Text.AlignLeft

Text.AlignHCenter

Text.AlignRight

Texto no editable (Text)

Es otro de los elementos básicos. Como características destacamos:

- Soporte para textos planos y ricos
- Soporte para marcas HTML
- Soporte para fuentes y estilos de la plataforma donde se ejecuta
- Alineación de texto: **verticalAlignment**, **horizontalAlignment**
- Multilínea

Texto editable de una sola línea (TextInput)

Entrada de texto de una solo línea. Como características destacamos:

- Soporte para fuentes y estilos de la plataforma donde se ejecuta
- Soporte para selección de texto
- Integración con el portapapeles
- Modo Echo para entrada de passwords
- Soporte para validadores de entrada: **IntValidator**, **DoubleValidator**, **RegExpValidator**

```
import QtQuick 1.0
TextInput {
    id: textInput
    text: "Hello world!"
    activeFocusOnPress: false
    MouseArea {
        anchors.fill: parent
        onClicked: {
            if (!textInput.activeFocus) {
                textInput.forceActiveFocus();
                textInput.openSoftwareInputPanel();
            } else {
                textInput.focus = false;
            }
        }
        onPressAndHold: textInput.closeSoftwareInputPanel();
    }
}
```

Texto editable multi-línea (TextEdit)

Entrada de texto de varias líneas. Como características destacamos:

Soporte para textos planos y ricos

- Muy similar a **Text** y **TextInput**
- Soporte para modo lectura (**readOnly**) y selección desde ratón (**selectByMouse**)
- Integración con el portapapeles



Otras lecturas relacionadas recomendadas:

- <http://doc.qt.nokia.com/4.7-snapshot/qml-item.html>