

QML et les styles

par traducteur : Thibaut Cuvelier ([Site Web](#)) ([Blog](#)) Qt Developer Network

Date de publication : 03/01/2011

Dernière mise à jour : 18/04/2011

QML fournit plusieurs mécanismes pour donner un style à une interface graphique. Voici trois approches communes.

Commentez !

I - L'article original.....	3
II - Les composants personnalisés.....	3
III - Un objet de style.....	3
IV - Une approche hybride.....	4

I - L'article original

Le Qt Developer Network est un réseau de développeurs utilisant Qt afin de partager leur savoir sur ce framework. **Vous pouvez le consulter en anglais.**

Nokia, Qt, Qt Quarterly et leurs logos sont des marques déposées de *Nokia Corporation* en Finlande et/ou dans les autres pays. Les autres marques déposées sont détenues par leurs propriétaires respectifs.

Cet article est la traduction de **Styling in QML**.

Merci à **ClaudeLELOUP** pour sa relecture orthographique !

II - Les composants personnalisés

QML supporte la définition de **nouveaux composants personnalisés**. Ci-dessous, on crée un composant personnalisé TitleText à utiliser à chaque fois qu'un titre est requis. Si on veut changer l'apparence de tous les titres dans l'interface, il suffit d'éditer ce fichier et les changements s'appliqueront à chaque fois que ce composant est utilisé.

TitleText.qml

```
Text {
    horizontalAlignment: Text.AlignHCenter
    font.pixelSize: 50
    color: "green"
}
```

On l'utilise comme ceci :

```
TitleText {
    text: "Title 1"
}
```

III - Un objet de style

Ici, on définit un objet Style qui contient une collection de propriétés définissant un style. Cet objet est instancié dans un composant racine, il est donc disponible dans toute l'application.

Style.qml

```
QObject {
    property int textSize: 20
    property color textColor: "green"
}
```

On l'active dans l'objet racine :

Composant racine

```
Rectangle {
    ...
    Style { id: style }
    ...
}
```

On l'utilise comme ceci :

```
Text {
    font.pixelSize: style.textSize
    color: style.textColor
}
```

```
text: "Hello World"
}
```

IV - Une approche hybride

On peut aussi définir un objet de style global et l'utiliser dans des composants personnalisés.

Style.qml

```
QtObject {
    property int titleAlignment: Text.AlignHCenter
    property int titleFontSize: 50
    property color titleColor: "green"
}
```

On l'active dans l'objet racine :

Composant racine

```
Rectangle {
    ...
    Style { id: style }
    ...
}
```

On définit un composant personnalisé utilisant ce style :

TitleText.qml

```
Text {
    horizontalAlignment: style.titleAlignment
    font.pixelSize: style.titleFontSize
    color: style.titleColor
}
```

On l'utilise comme ceci :

```
TitleText {
    text: "Title 1"
}
```