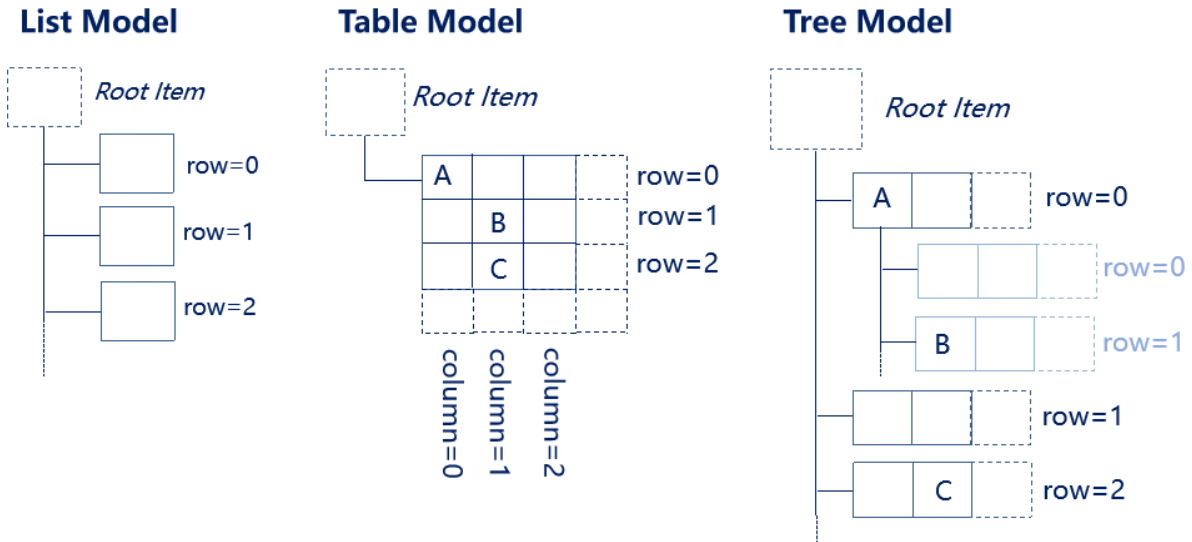


48 MV相关概念

48.1 数据模型的基本结构

在Qt中，无论底层数据的存储结构如何，模型总是以列表（List）、表格（Table）或树（Tree），这三种基本结构中的一种表示数据，如下图所示：



它们被分别称为列表模型（List Model）、表格模型（Table Model）和树模型（Tree Model）。模型中数据的基本单位是项（Item），每个项都有一个行号、一个列号和一个父项。在列表和表格模型中，所有项都有一个相同的父项，而在树模型中，行号、列号和父项的情况则会复杂一些，但借助这三个坐标，依然可以唯一确定一个项的位置，并存取其中的数据。

48.2 模型索引

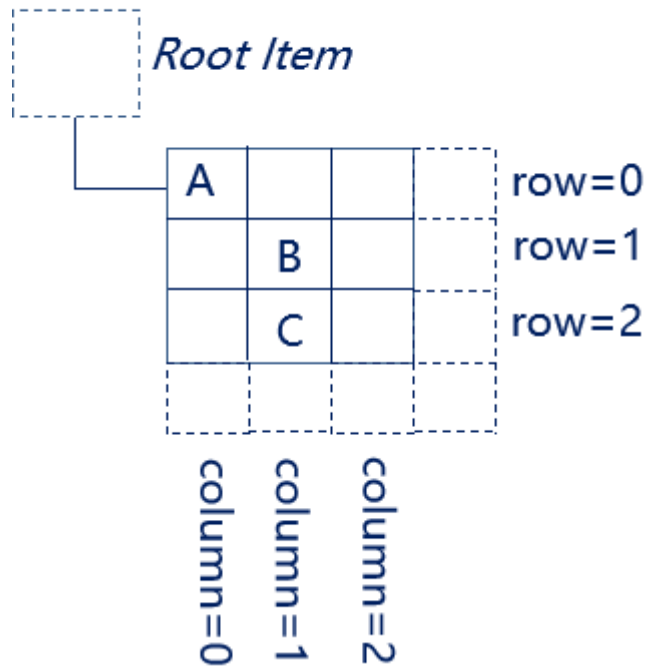
为了将数据的表示方式与存储方式隔离，模型引入了索引的概念。通过模型存取的每个数据项都有唯一的模型索引与之对应。视图和代理都通过模型索引访问模型中的数据。

QModelIndex是表示模型索引的类。模型索引相当于一个指向模型中数据项的临时指针，视图和代理通过它访问模型中的数据。之所以称之为临时指针，是因为模型中的数据组织结构随时可能发生变化，此刻的索引在未来未必依然有效。如果想获得持久性模型索引，可以使用QPersistentModelIndex类。

48.3 行号、列号和父项

要获得一个模型索引，必须提供三个参数，即行号、列号和父项。它们是在模型中定位某个具体数据项的三个坐标。例如在下面的表格模型中：

Table Model



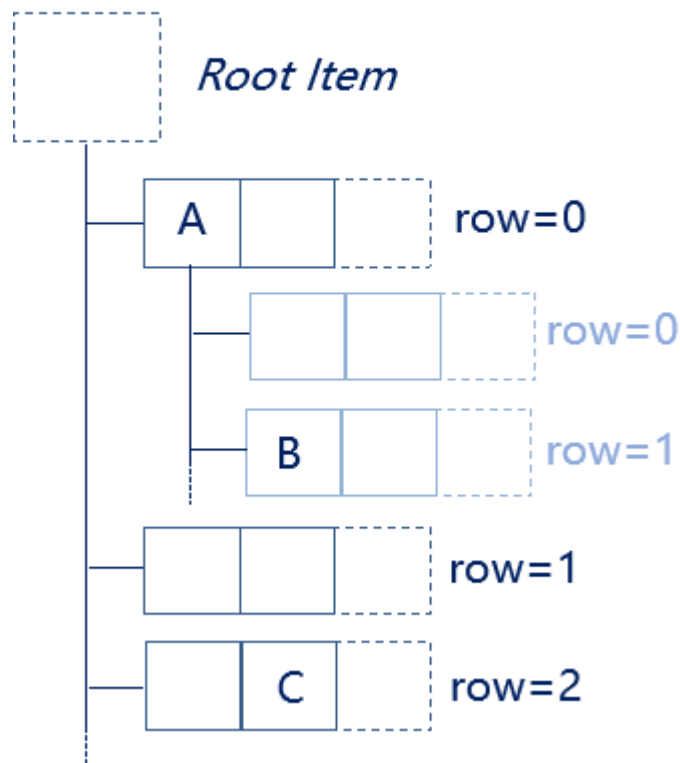
数据项A、B、C的模型索引为：

```
1 QModelIndex indexA = model->index(0, 0, QModelIndex());  
2 QModelIndex indexB = model->index(1, 1, QModelIndex());  
3 QModelIndex indexC = model->index(2, 1, QModelIndex());
```

其中，QModelIndex()表示根项（Root Item）的模型索引。A、B、C三个数据项的父项都是根项。

又比如在下方的树模型中：

Tree Model



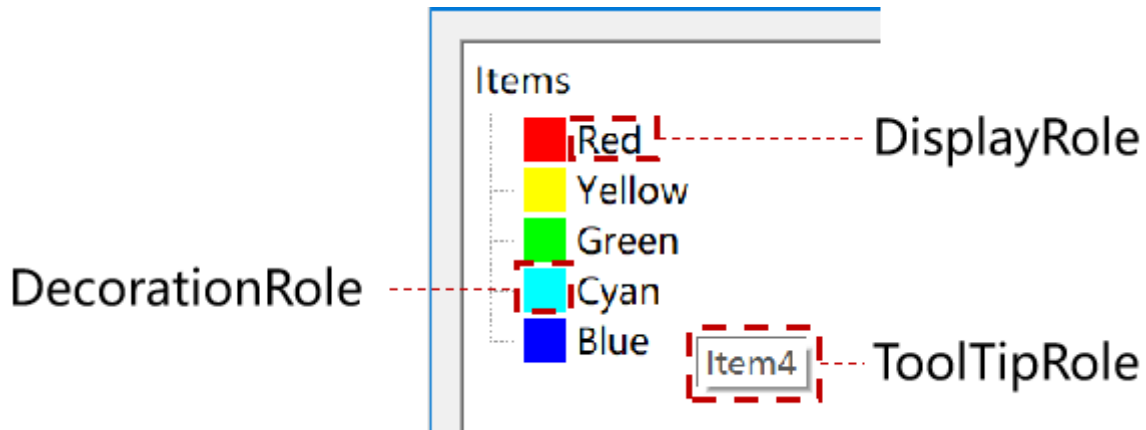
数据项A、B、C的模型索引为：

```
1 QModelIndex indexA = model->index(0, 0, QModelIndex());
2 QModelIndex indexB = model->index(1, 0, indexA);
3 QModelIndex indexC = model->index(2, 1, QModelIndex());
```

注意，数据项B的父项是A项而非根项。

48.4 数据角色

模型中的每个项可以包含多个扮演不同角色的数据，它们用于控制视图和代理显示数据的方式。例如下图中的数据项：



- 装饰角色 (DecorationRole) 的数据表示项的颜色、图标或图片
- 显示角色 (DisplayRole) 的数据表示项的文本
- 提示角色 (ToolTipRole) 的数据表示鼠标停留于该项时显示的提示信息

在为模型中的项设置数据时，需要指定其角色。例如，标准项模型 (QStandardItemModel) 中的项用 QStandardItem 类表示，为其设置数据的方法是：

```
1 void QStandardItem::setData(const QVariant& value, int role = Qt::UserRole + 1);
```

其中，value 参数表示数据的值，role 参数表示数据的角色。

数据的角色用 Qt::ItemDataRole 枚举类型表示，可取以下值：

- Qt::DecorationRole: 装饰角色
- Qt::DisplayRole: 显示角色
- Qt::ToolTipRole: 提示角色
- Qt::UserRole: 自定义角色

在从项中获取数据时，同样需要指定角色，以明确究竟需要获取哪种角色的数据：

```
1 QVariant QStandardItem::data(int role = Qt::UserRole + 1) const;
```

不同的视图对数据角色的解释和显示可能并不相同，甚至可能忽略某些角色的数据。