# Qt解析XML文件

QtXml模块提供了一个读写XML文件的流，解析方法包含DOM和SAX,两者的区别是什么呢？

* DOM（Document Object Model）：将XML文件保存为树的形式，操作简单，便于访问。
* SAX（Simple API for XML）：接近于底层，速度较快，但不便于访问。

例如：

person.xml:



解析方法：

void ParseXML::parse(QString file\_name)

{

    if(file\_name.isEmpty())

        return;

    QFile file(file\_name);

    if(!file.open(QFile::ReadOnly | QFile::Text))

    {

        QMessageBox::information(NULL, QString("title"), QString("open error!"));

        return;

    }

    QDomDocument document;

    QString error;

    int row = 0, column = 0;

    if(!document.setContent(&file, false, &error, &row, &column))

    {

        QMessageBox::information(NULL, QString("title"), QString("parse file failed at line row and column") + QString::number(row, 10) + QString(",") + QString::number(column, 10));

        return;

    }

    if(document.isNull())

    {

        QMessageBox::information(NULL, QString("title"), QString("document is null!"));

        return;

    }

    QDomElement root = document.documentElement();

    //root\_tag\_name为persons

    QString root\_tag\_name = root.tagName();

    if(root.hasAttribute("name"))

    {

        //name为Qt

        QString name = root.attributeNode("name").value();

    }

    //获取id="1"的节点

    QDomElement person = root.firstChildElement();

    if(person.isNull())

        return;

    QString person\_tag\_name = person.tagName();

    //id为1

    QString id = person.attributeNode("id").value();

    //获取子节点，数目为2

    QDomNodeList list = root.childNodes();

    int count = list.count();

    for(int i=0; i

    {

        QDomNode dom\_node = list.item(i);

        QDomElement element = dom\_node.toElement();

        //获取id值，等价

        QString id\_1 = element.attributeNode("id").value();

        QString id\_2 = element.attribute("id");

        //获取子节点，数目为4，包括：name、age、email、website

        QDomNodeList child\_list = element.childNodes();

        int child\_count = child\_list.count();

        for(int j=0; j

        {

            QDomNode child\_dom\_node = child\_list.item(j);

            QDomElement child\_element = child\_dom\_node.toElement();

            QString child\_tag\_name = child\_element.tagName();

            QString child\_\_tag\_value = child\_element.text();

        }

    }

    //按照name、age、email、website的顺序获取值

    QDomElement element = person.firstChildElement();

    while(!element.isNull())

    {

        QString tag\_name = element.tagName();

        QString tag\_value = element.text();

        element = element.nextSiblingElement();

    }

}