



母狮惨遭水牛顶伤



抓拍鲨鱼呲牙怪笑照



曼妙旅程:潜拍海底世界



嬉戏野兔击掌求偶

[查看更多>>](#)

谁看过这篇博文

	t201106	3月2日
	Gentleman	3月2日
	木船	3月1日
	花丿开、…	2月27日
	spielmanlee	2月26日
	moonlqer	2月26日
	wdcjq2028	2月26日
	右右youyou	2月24日
	蓝风	2月23日
	today sky	2月22日
	Geeprox	2月21日
	jiayuan_x…	2月14日

```
{
    return;
}

QStringList filters;
filters<<QString("*.jpeg")<<QString("*.jpg")<<QString("*.png")<<QString("*.tiff")
<<QString("*.gif")<<QString("*.bmp");
dir.setFilter(QDir::Files | QDir::NoSymLinks); //设置类型过滤器，只为文件格式
dir.setNameFilters(filters); //设置文件名称过滤器，只为filters格式（后缀为.jpeg等图片格式）

int dir_count = dir.count();
if(dir_count <= 0)
{
    return;
}

QStringList string_list;
//获取分隔符
//QChar separator = QDir::separator();
QChar separator = QChar('/');
if(!path.contains(separator))
{
    separator = QChar('\\');
}
QChar last_char = path.at(path.length()-1);
if(last_char == separator)
{
    separator = QChar();
}

for(uint i=0; i
{
    QString file_name = dir[i]; //文件名称
    QString file_path = path + separator + file_name; //文件全路径
    string_list.append(file_path);
}

//string_list 添加完成之后，就可以查看list中的文件路径了
}
```

分析：setNameFilters顾名思义，就是过滤文件名称的。如果只需要获取指定路径下的文件名，则可去掉“获取分隔符”部分代码（因为我是为了获取文件的全路径）。

思考：QDir::separator() 这是用于获取分隔符的，调试过程中发现path的分隔符为'/'，奇怪的是获取到的为'\'，刚好相反，所以我通过contains的方式获取分隔符的（无非'/'与'\''）。

讨论：如果设定filters后，那么存在一定的问题。熟用Windows的应该都知道，文件名是忽略大小写的（包括扩展名），那么若filters设定了“*.jpg”之后，则就不可添加扩展名为“.JPG”、“.Jpg”等大小写兼有的文件了。

既然有问题，就有解决问题的方式。

- 1、问题来源是由扩展名引起，那么去掉setNameFilters(filters);
- 2、上述已经获取文件全路径，那么QFileInfo file_info (file_path) 获取文件信息的对象
- 3、通过file_info.suffix() 或者completeSuffix() 来判定文件的后缀、扩展名
- 4、获取之后比较时忽略大小写即可。如：QString::compare(suffix, QString("*.jpeg"), Qt::CaseInsensitive) == 0

总结：通过以上两种方式比较，关于遍历下一层的方式，采用“方式1”较好。

二、级联遍历文件夹及其子孙文件夹中的文件

方式1:

```
void ImageTree::addSubFolderImages(QString path)
{
    //判断路径是否存在
    QDir dir(path);
    if(!dir.exists())
    {
        return;
    }
}
```

```
//获取所选文件类型过滤器
QStringList filters;
filters<<QString("*.jpeg")<<QString("*.jpg")<<QString("*.png")<<QString("*.tiff")
<<QString("*.gif")<<QString("*.bmp");
```

```
//定义迭代器并设置过滤器
QDirIterator dir_iterator(path,
    filters,
    QDir::Files | QDir::NoSymLinks,
    QDirIterator::Subdirectories);
QStringList string_list;
while(dir_iterator.hasNext())
{
    dir_iterator.next();
    QFileInfo file_info = dir_iterator.fileInfo();
    QString absolute_file_path = file_info.absoluteFilePath();
    string_list.append(file_path);
}
}
```

分析：QDirIterator定义过程中可设置过滤器，包括：文件名称、文件类型等。dir_iterator.next()这句话很重要，如果缺少将会进入死循环！

方式2:

```
QStringList string_list;
void ImageTree::addSubFolderImages(QString path)
{
    QDir dir(path);
    if(!dir.exists())
    {
        return;
    }
    dir.setFilter(QDir::Dirs | QDir::Files | QDir::NoSymLinks);
    dir.setSorting(QDir::DirsFirst);
    QFileInfoList list = dir.entryInfoList();
    int i = 0;
    bool is_dir;
    do
    {
        QFileInfo file_info = list.at(i);
        if(file_info.fileName() == "." | file_info.fileName() == "..")
        {
            i++;
            continue;
        }

        is_dir = file_info.isDir();
        if(is_dir)
        {
            //进行递归
            addSubFolderImages(file_info.filePath());
        }
        else
        {
            //获取文件后缀并获取所选包含类型，若存在包含类型且后缀相同，则添加
            QString suffix = file_info.suffix();
            if(QString::compare(suffix, QString("png"), Qt::CaseInsensitive) == 0)
            {
                QString absolute_file_path = file_info.absoluteFilePath();
                string_list.append(absolute_file_path);
            }
        }
        i++;
    }
    while(i
```

分析：此方式采用递归的思路解决，也是网上大多数人用的办法，个人建议摒弃！递归的效率真心不敢接受，而且代码看起来也费劲。

总结：通过以上两种方式比较，关于遍历子孙文件夹的方式，采用“方式1”较好。

关于QFileInfo获取文件信息的方法可以查看API

方法如下：

```
void setFile(const QString &file);
void setFile(const QFile &file);
void setFile(const QDir &dir, const QString &file);
bool exists() const;
void refresh();

QString filePath() const;
QString absoluteFilePath() const;
QString canonicalFilePath() const;
QString fileName() const;
QString baseName() const;
QString completeBaseName() const;
QString suffix() const;
QString bundleName() const;
QString completeSuffix() const;

QString path() const;
QString absolutePath() const;
QString canonicalPath() const;
QDir dir() const;
QDir absoluteDir() const;

bool isReadable() const;
bool isWritable() const;
bool isExecutable() const;
bool isHidden() const;
bool isNativePath() const;

bool isRelative() const;
inline bool isAbsolute() const { return !isRelative(); }
bool makeAbsolute();

bool isFile() const;
bool isDir() const;
bool isSymLink() const;
bool isRoot() const;
bool isBundle() const;

QString readLink() const;
inline QString symLinkTarget() const { return readLink(); }

QString owner() const;
uint ownerId() const;
QString group() const;
uint groupId() const;

bool permission(QFile::Permissions permissions) const;
QFile::Permissions permissions() const;

qint64 size() const;

QDateTime created() const;
QDateTime lastModified() const;
QDateTime lastRead() const;

bool caching() const;
void setCaching(bool on);
```

更多关于获取文件图标、类型的详细信息请查看：

- Qt之QFileIconProvider（获取文件图标、类型）。
- Qt之QFileIconProvider续（获取文件图标、类型）。

注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者：`☆奋斗ing♥孩子` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90101f1ur.html。

5

👍喜欢

分享：       

阅读(5714) | 评论 (0) | 收藏(0) | 已有7人转载▼ | 喜欢▼ | 打印

已投稿到： 排行榜

前一篇：Qt之自定义菜单（按钮菜单）
后一篇：Qt技术交流群（26197884、26188347）

评论

重要提示：警惕虚假中奖信息

[发评论]

做第一个评论者吧！ 抢沙发>>

发评论




☐  分享到微博 

☐ 匿名评论

验证码： [请点击后输入验证码](#) [收听验证码](#)

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇
Qt之自定义菜单（按钮菜单）

后一篇 >
Qt技术交流群（26197884、26188347）

