# File类

## 概述

File更应该叫做一个路径，文件路径或者文件夹路径



path 路径：绝对路径："D:\\io\\helloworld.txt"

相对路径："helloworld.txt"

## 构造方法

获取文件本身的一些信息，不涉及对文件的读写操作。文件目录和文件。

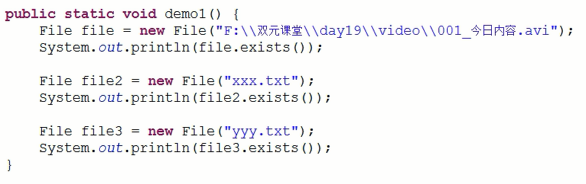
File(String pathname)：根据一个路径得到File对象

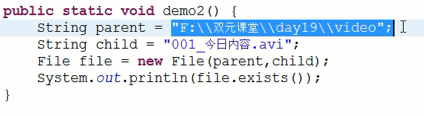
File(String parent, String child):根据一个目录和一个子文件/目录得到File对象

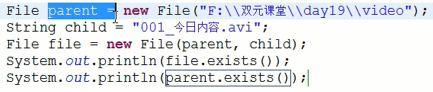
File(File parent, String child):根据一个父File对象和一个子文件/目录得到File对象

父目录 + 子文件

[File](mk:@MSITStore:C:\Users\Administrator.LS--20161013LKV\Desktop\参考API\JDK_API_1_6_zh_CN.CHM::/java/io/File.html#File(java.net.URI))([URI](mk:@MSITStore:C:\Users\Administrator.LS--20161013LKV\Desktop\参考API\JDK_API_1_6_zh_CN.CHM::/java/net/URI.html) uri) 相对路径：







## 常用方法

### 创建功能

public boolean createNewFile(): 创建文件 如果存在这样的文件，就不创建了

public boolean mkdir(): 创建文件夹 如果存在这样的文件夹，就不创建了

public boolean mkdirs(): 创建文件夹,如果父文件夹不存在，会帮你创建出来

### 重命名和删除功能

public boolean renameTo(File dest):把文件重命名为指定的文件路径

重命名注意事项

如果路径名相同，就是改名。

如果路径名不同，就是改名并剪切。

public boolean delete():删除文件或者文件夹

删除注意事项：

Java中的删除不走回收站。

要删除一个文件夹，请注意该文件夹内不能包含文件或者文件夹

### 判断功能

public boolean isDirectory():判断是否是目录

public boolean isFile():判断是否是文件

public boolean exists():判断是否存在

public boolean canRead():判断是否可读

public boolean canWrite():判断是否可写

public boolean isHidden():判断是否隐藏

### 获取功能

public String getAbsolutePath()：获取绝对路径

public String getPath(): 获取构造方法中传入路径

public String getName(): 获取名称

public long length(): 获取长度。字节数

public long lastModified(): 获取最后一次的修改时间，毫秒值

public String[] list(): 获取指定目录下的所有文件或者文件夹的名称数组

public File[] listFiles(): 获取指定目录下的所有文件或者文件夹的File数组

# 递归

**public** **class** Demo8\_Digui {

/\*\*

\* **@param** args

\* 递归:方法自己调用自己

\* 5!

\* 5 \* 4 \* 3 \* 2 \* 1

\*

\* 5 \* fun(4)(代表4!)

\* 4 \* fun(3)(代表3!)

\* 3 \* fun(2)(代表2!)

\* 2 \* fun(1)(代表1!)

\* 递归的弊端:不能调用次数过多,容易导致栈内存溢出

\* 递归的好处:不用知道循环次数

\*

\* 构造方法是否可以递归调用?

\* 构造方法不能使用递归调用

\*

\* 递归调用是否必须有返回值?

\* 不一定(可以有,也可以没有)

\*/

**public** **static** **void** main(String[] args) {

System.*out*.println(*fun*(5));

}

**public** **static** **int** fun(**int** num) {

**if**(num == 1) {

**return** 1;

}**else** {

**return** num \* *fun*(num - 1);

}

}

}

**public** **class** Test5 {

/\*\*

\* 需求:从键盘输入接收一个文件夹路径,打印出该文件夹下所有的.java文件名

\*

\* 分析:

\* 从键盘接收一个文件夹路径

\* 1,如果录入的是不存在,给与提示

\* 2,如果录入的是文件路径,给与提示

\* 3,如果是文件夹路径,直接返回

\*

\* 打印出该文件夹下所有的.java文件名

\* 1,获取到该文件夹路径下的所有的文件和文件夹,存储在File数组中

\* 2,遍历数组,对每一个文件或文件夹做判断

\* 3,如果是文件,并且后缀是.java的,就打印

\* 4,如果是文件夹,就递归调用

\*/

**public** **static** **void** main(String[] args) {

File dir = *getDir*();

*printJavaFile*(dir);

}

/\*

\* 获取键盘录入的文件夹路径

\* 1,返回值类型File

\* 2,不需要有参数

\*/

**public** **static** File getDir() {

Scanner sc = **new** Scanner(System.*in*); //创建键盘录入对象

System.*out*.println("请输入一个文件夹路径");

**while**(**true**) {

String line = sc.nextLine(); //将键盘录入的文件夹路径存储

File dir = **new** File(line); //封装成File对象

**if**(!dir.exists()) {

System.*out*.println("您录入的文件夹路径不存在,请重新录入");

}**else** **if**(dir.isFile()) {

System.*out*.println("您录入的是文件路径,请重新录入文件夹路径");

}**else** {

**return** dir;

}

}

}

/\*

\* 获取文件夹路径下的所.java文件

\* 1,返回值类型 void

\* 2,参数列表File dir

\*/

**public** **static** **void** printJavaFile(File dir) {

//1,获取到该文件夹路径下的所有的文件和文件夹,存储在File数组中

File[] subFiles = dir.listFiles();

//2,遍历数组,对每一个文件或文件夹做判断

**for** (File subFile : subFiles) {

//3,如果是文件,并且后缀是.java的,就打印

**if**(subFile.isFile() && subFile.getName().endsWith(".java")) {

System.*out*.println(subFile);

//4,如果是文件夹,就递归调用

}**else** **if** (subFile.isDirectory()){

*printJavaFile*(subFile);

}

}

}

}

# 案例

## 判断E盘目录下是否有后缀名为.jpg的文件

需求：判断E盘目录下是否有后缀名为.jpg的文件，如果有，就输出该文件名称

**public** **static** **void** main(String[] args) {

File dir = **new** File("E:\\");

File[] subFiles = dir.listFiles();//获取 所有文件或文件夹

**for** (File subFile : subFiles) {

**if** (subFile.isFile() && subFile.getName().endsWith(".jpg")) {

System.*out*.println(subFile);

}

}

}

## 文件名称过滤器