24 File

- **路径：**
  - **相对路径和绝对路径**

    相对路径

    路径1："a.txt"
    路径2："abc\\a.txt"

    绝对路径

    路径1："C:\\a.txt"
    路径2："C:\\abc\\a.txt"

  - 绝对路径是带盘符的。
  - 相对路径是不带盘符的，默认到当前项目下去找。
  - **父级路径和子级路径**
    - 例：C:\\Users\\QHN\\Desktop\\a.txt
    - 父级：C:\\Users\\QHN\\Desktop
    - 子级：a.txt
- **File类相关内容**
  - File对象就表示一个路径，可以是文件的路径、也可以是文件夹的路径
  - 这个路径可以是存在的，也允许是不存在的

方法名称	说明
public File(String pathname)	根据文件路径创建文件对象
public File(String parent, String child)	根据父路径名字符串和子路径名字符串创建文件对象
public File(File parent, String child)	根据父路径对应文件对象和子路径名字符串创建文件对象

```java
//1.根据字符串表示的路径，变成File对象
String str = "C:\\Users\\alienware\\Desktop\\a.txt";
File f1 = new File(str);
System.out.println(f1);//C:\Users\alienware\Desktop\a.txt

//2.父级路径：C:\Users\alienware\Desktop
//子级路径：a.txt
String parent = "C:\\Users\\alienware\\Desktop";
String child = "a.txt";
File f2 = new File(parent,child);
System.out.println(f2);//C:\Users\alienware\Desktop\a.txt

File f3 = new File( pathname: parent + "\\" + child);
System.out.println(f3);//C:\Users\alienware\Desktop\a.txt


//3.把一个File表示的路径和String表示路径进行拼接
File parent2 = new File( pathname: "C:\\Users\\alienware\\Desktop");
String child2 = "a.txt";
File f4 = new File(parent2,child2);
System.out.println(f4);//C:\Users\alienware\Desktop\a.txt
```

- **File的常见成员方法**

方法名称	说明
public boolean isDirectory()	判断此路径名表示的File是否为文件夹
public boolean isFile()	判断此路径名表示的File是否为文件
public boolean exists()	判断此路径名表示的File是否存在
public long length()	返回文件的大小（字节数量）
public String getAbsolutePath()	返回文件的绝对路径
public String getPath()	返回定义文件时使用的路径
public String getName()	返回文件的名称，带后缀
public long lastModified()	返回文件的最后修改时间（时间毫秒值）

方法名称	说明
public boolean createNewFile()	创建一个新的空的文件
public boolean mkdir()	创建单级文件夹
public boolean mkdirs()	创建多级文件夹
public boolean delete()	删除文件、空文件夹

delete方法默认只能删除文件和空文件夹，delete方法直接删除不走回收站

方法名称	说明
public File[] listFiles()	获取当前该路径下所有内容

- 当调用者File表示的路径不存在时，返回null
- 当调用者File表示的路径是文件时，返回null
- 当调用者File表示的路径是一个空文件夹时，返回一个长度为0的数组
- 当调用者File表示的路径是一个有内容的文件夹时，将里面所有文件和文件夹的路径放在File数组中返回
- 当调用者File表示的路径是一个有隐藏文件的文件夹时，将里面所有文件和文件夹的路径放在File数组中返回，包含隐藏文件
- 当调用者File表示的路径是需要权限才能访问的文件夹时，返回null

方法名称	说明
public static File[] listRoots()	列出可用的文件系统根
public String[] list()	获取当前该路径下所有内容
public String[] list(FilenameFilter filter)	利用文件名过滤器获取当前该路径下所有内容
public File[] listFiles()	获取当前该路径下所有内容
public File[] listFiles(FileFilter filter)	利用文件名过滤器获取当前该路径下所有内容
public File[] listFiles(FilenameFilter filter)	利用文件名过滤器获取当前该路径下所有内容

- **1. 获取**

```java
//1.对一个文件的路径进行判断
File f1 = new File( pathname: "D:\\aaa\\a.txt");
System.out.println(f1.isDirectory());//false
System.out.println(f1.isFile());//true
System.out.println(f1.exists());//true
System.out.println("-------------------------------------");
//2.对一个文件夹的路径进行判断
File f2 = new File( pathname: "D:\\aaa\\bbb");
System.out.println(f2.isDirectory());//true
System.out.println(f2.isFile());//false
System.out.println(f2.exists());//true
System.out.println("-------------------------------------");
//3.对一个不存在的路径进行判断
File f3 = new File( pathname: "D:\\aaa\\c.txt");
System.out.println(f3.isDirectory());//false
System.out.println(f3.isFile());//false
System.out.println(f3.exists());//false
```

- **2. 判断**

```java
//1.length  返回文件的大小（字节数量）
//细节1：这个方法只能获取文件的大小，单位是字节
//如果单位我们要是M，G，可以不断的除以1024
//细节2：这个方法无法获取文件夹的大小
//如果我们要获取一个文件夹的大小，需要把这个文件夹里面所有的文件大小都累加在一起。

File f1 = new File( pathname: "D:\\aaa\\a.txt");
long len = f1.length();
System.out.println(len);//12

File f2 = new File( pathname: "D:\\aaa\\bbb");
long len2 = f2.length();
System.out.println(len2);//0
```

```java
//2.getAbsolutePath 返回文件的绝对路径
File f3 = new File( pathname: "D:\\aaa\\a.txt");
String path1 = f3.getAbsolutePath();
System.out.println(path1);

File f4 = new File( pathname: "myFile\\a.txt");
String path2 = f4.getAbsolutePath();
System.out.println(path2);
```

```java
//3.getPath 返回定义文件时使用的路径
File f5 = new File( pathname: "D:\\aaa\\a.txt");
String path3 = f5.getPath();
System.out.println(path3);//D:\aaa\a.txt

File f6 = new File( pathname: "myFile\\a.txt");
String path4 = f6.getPath();
System.out.println(path4);//myFile\a.txt

//4.getName 获取名字
//细节1:
//a.txt:
//      a 文件名
//      txt 后缀名、扩展名
//细节2:
//文件夹：返回的就是文件夹的名字
File f7 = new File( pathname: "D:\\aaa\\a.txt");
String name1 = f7.getName();
System.out.println(name1);


File f8 = new File( pathname: "D:\\aaa\\bbb");
String name2 = f8.getName();
System.out.println(name2);//bbb

//5.lastModified  返回文件的最后修改时间（时间毫秒值）
File f9 = new File( pathname: "D:\\aaa\\a.txt");
long time = f9.lastModified();
System.out.println(time);//1667380952425
```

- getPath：括号里面是什么路径，就返回什么路径

- **3.创建**

```java
//1.createNewFile 创建一个新的空的文件
//细节1：如果当前路径表示的文件是不存在的，则创建成功，方法返回true
//      如果当前路径表示的文件是存在的，则创建失败，方法返回false
//细节2：如果父级路径是不存在的，那么方法会有异常IOException
//细节3：createNewFile方法创建的一定是文件，如果路径中不包含后缀名，则创建一个没有后缀的文件
File f1 = new File( pathname: "D:\\aaa\\ddd");
boolean b = f1.createNewFile();
System.out.println(b);//true
```

```java
//2.mkdir   make Directory，文件夹（目录）
//细节1：windows当中路径是唯一的，如果当前路径已经存在，则创建失败，返回false
//细节2：mkdir方法只能创建单级文件夹，无法创建多级文件夹。
File f2 = new File( pathname: "D:\\aaa\\aaa\\bbb\\ccc");
boolean b = f2.mkdir();
System.out.println(b);

//3.mkdirs   创建多级文件夹
//细节：既可以创建单级的，又可以创建多级的文件夹
File f3 = new File( pathname: "D:\\aaa\\ggg");
boolean b = f3.mkdirs();
System.out.println(b);//true
```

- **4.删除**

```java
细节：
    如果删除的是文件，则直接删除，不走回收站。
    如果删除的是空文件夹，则直接删除，不走回收站
    如果删除的是有内容的文件夹，则删除失败
*/


//1.创建File对象
File f1 = new File( pathname: "D:\\aaa\\eee");
//2.删除
boolean b = f1.delete();
System.out.println(b);
```

- **5.读取并遍历**

```java
//1.创建File对象
File f = new File( pathname: "D:\\aaa");
//2.listFiles方法
//作用：获取aaa文件夹里面的所有内容，把所有的内容放到数组中返回
File[] files = f.listFiles();
for (File file : files) {
    //file依次表示aaa文件夹里面的每一个文件或者文件夹
    System.out.println(file);
}
```