# 文件系统

是用php去操作磁盘上的文件

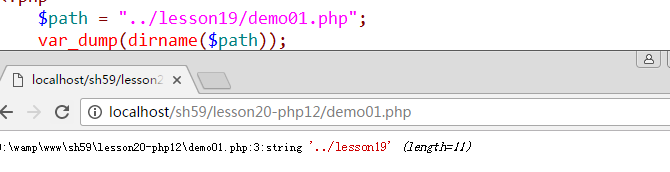
文件类型：file文件，dir目录，unknown位置类型

File：表示是可以用于保存数据的一个文件类型。根据文件的后缀名来区分

Dir：目录，给文件归类。

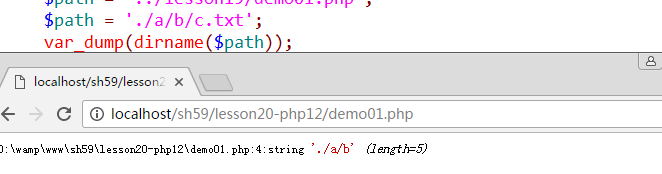
# 路径的基本操作

Dirname():获取路径字符串的目录部分

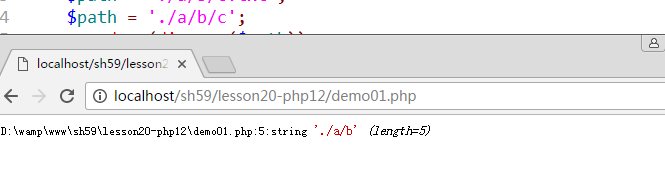


参数：字符串的路径

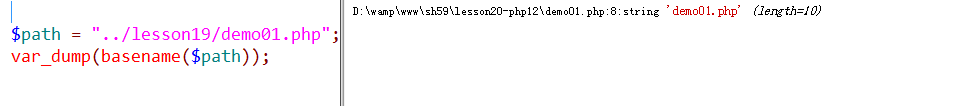
只要满足路径的格式，即可获取到



获取的是最后一个 / 的前面的那部分

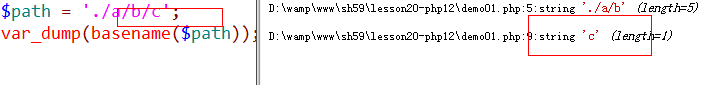


Basename():获取路径中文件名部分



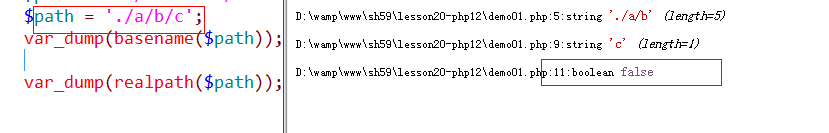
获取一个字符串路径的文件部分

是获取 最后一个/ 的部分

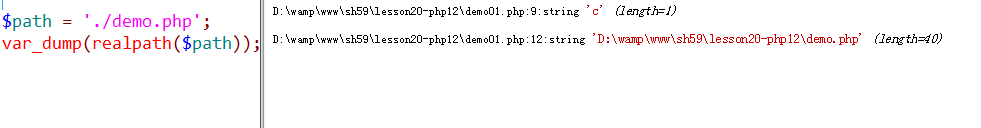


Realpath():表示获取真实存在磁盘上路径的对应的一个绝对路径

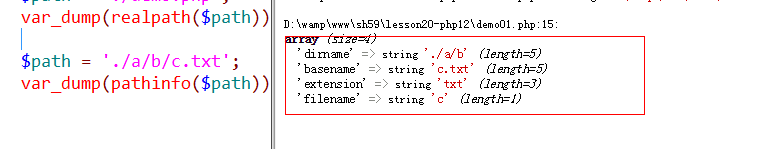
如果路径不存在返回false



如果存在返回绝对路径



Pathinfo():获取当前文件的路径的信息



第一个参数：当前的路径信息

返回一个数组类型的数据

Dirname:返回目录部分

Basename：返回的是除了目录以外的部分

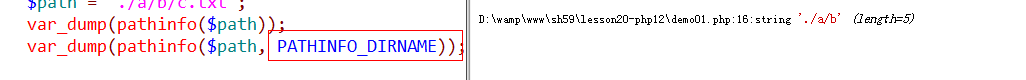
Extension：返回的当前路径后缀名

Filename：当前文件的名称

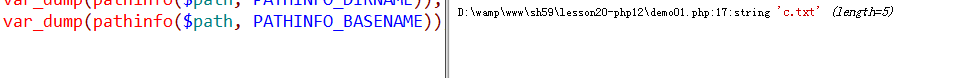
第二个参数：

获取四个其中的任意一个部分

PATHINFO\_DIRNAME:

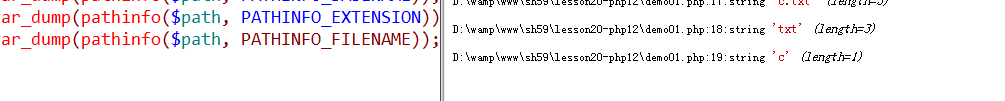


PATHINFO\_BASENAME



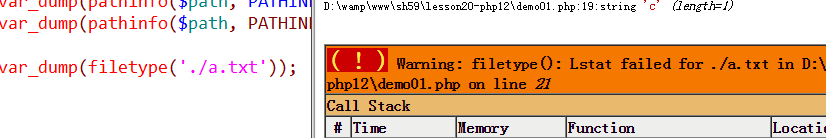
PATHINFO\_EXTENSION

PATHINFO\_FILENAME

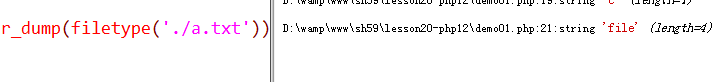


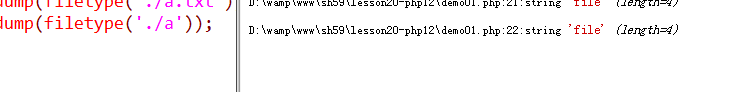
# 判断文件的类型

Filetype():获取文件的类型



必须保持文件存在于当前的磁盘上



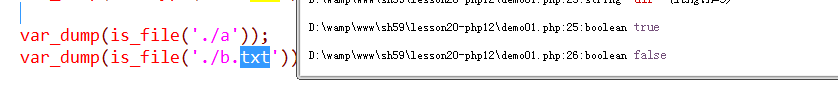


根据文件的后缀名没有任何的关系



Is\_file():判断是否是一个file类型

是file类型，返回true，否则返回false

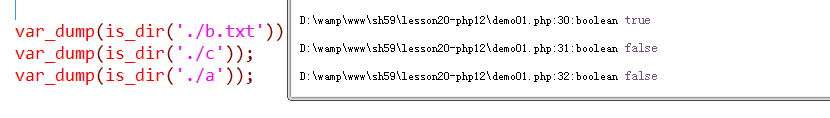


如果文件不存在返回false



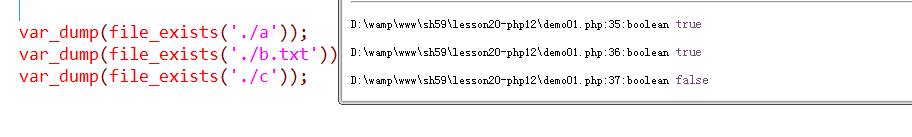
Is\_dir():判断是否一个dir类型

如果是返回true，否则反false



# 判断文件是否存在

File\_exists():判断文件是否存在file，dir

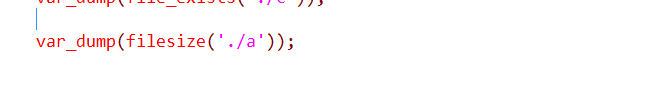


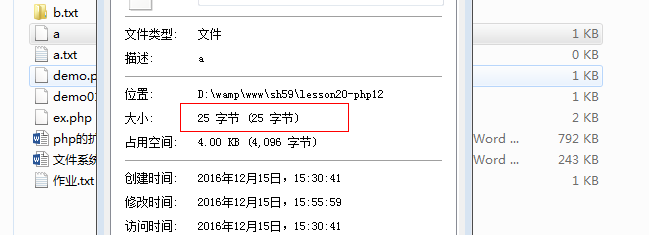
# 获取的文件的大小

只能去获取file类型大小，win系统的下文件的字节数

Filesize()

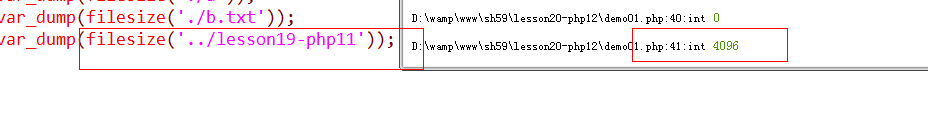
参数：文件路径





获取dir类型大小





# 获取文件的时间

Win下面的时间，是以时间戳的形式表示

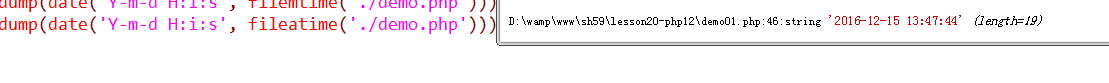
Filectime():获取文件的创建时间



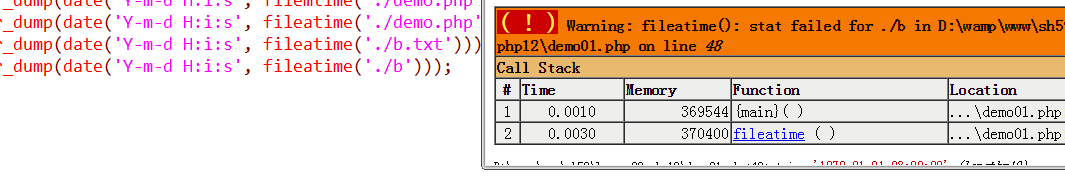
Filemtime():获取文件的修改时间



Fileatime(): 获取文件的访问时间



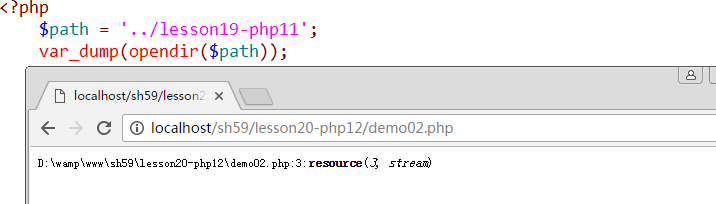
如果文件不存在则会报错



# 读取目录中文件

## 打开目录

Opendir():打磁盘上的目录



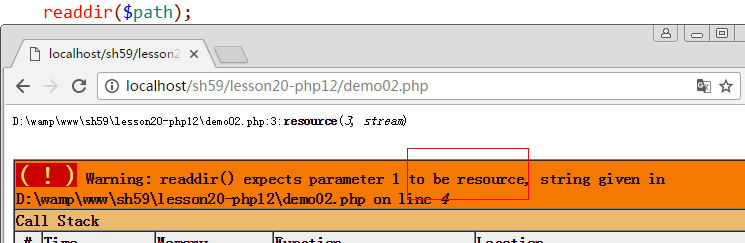
参数：目录的所在位置

返回值：资源类型的数据

## 读取目录

Readdir():读取目录中的文件名

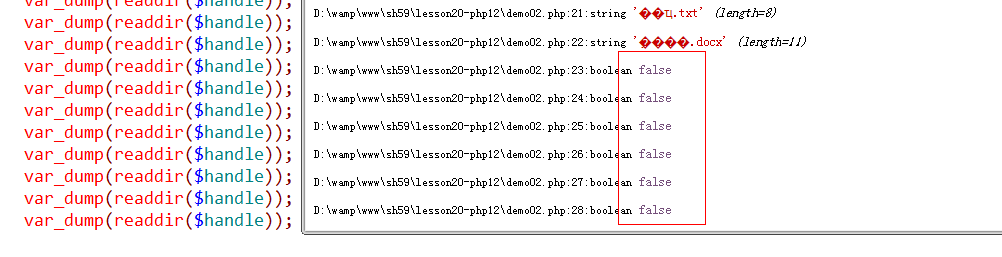
参数：使用opendir打开的目录的资源



在每一目录中，都存在一个 . 和 ..

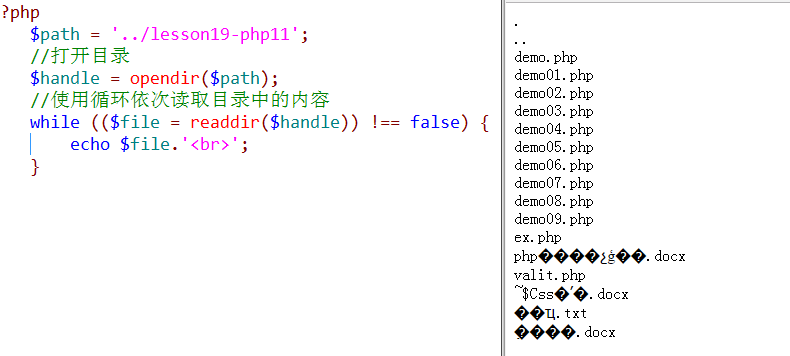


读取的文件的原理，就是使用指针来操作，当指针非法的时候，返回false

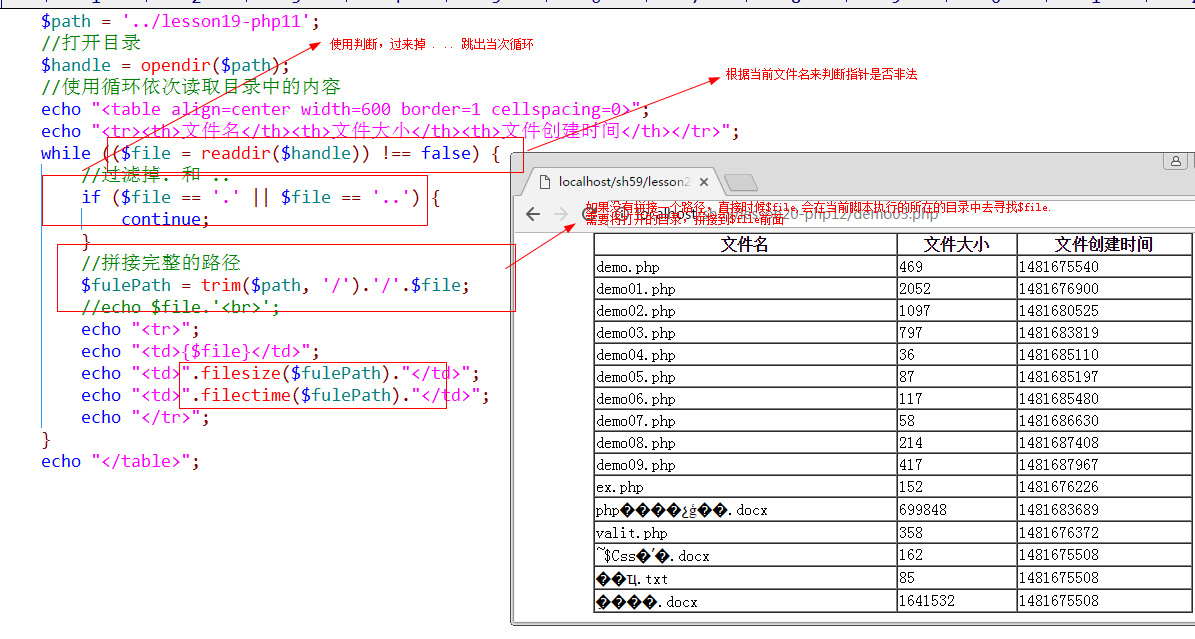


## 使用while读取文件

1、简单的读取文件名



2、获取文件的基本信息，文件的大小，文的创建时间



## 递归遍历目录

递归入口：什么时候需要执行递归

递归出口：什么时候递归可以结束

1. 递归的函数遍历目录的时候

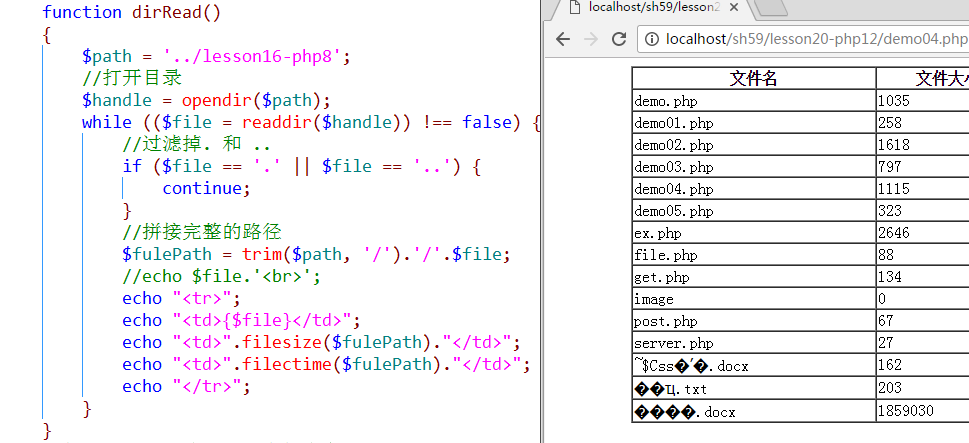
入口：如果是目录，执行递归

出口：文件如果不是目录

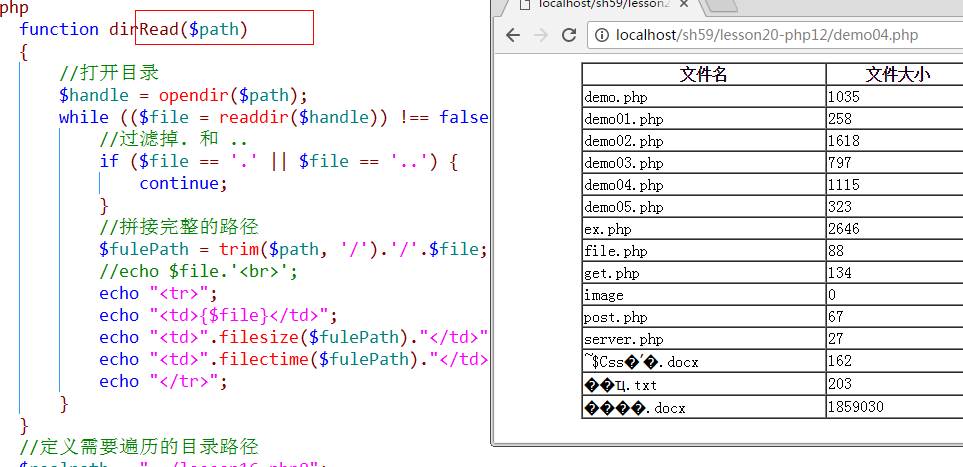
Is\_dir()

Is\_file()

将函数的功能实现了，将直接将功能封装在函数里面



1. 需要根据不同的需要，去改变函数里面的一些参数



改变了目录路径，可以变量任何一个目录

1. 判断当前的文件的类型是否是否一个dir



1. 目录都有层级之分。

向在页面以树状图的形式表现出目录的结构。

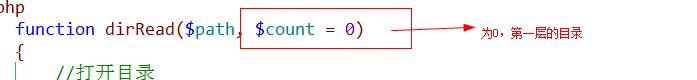
|--

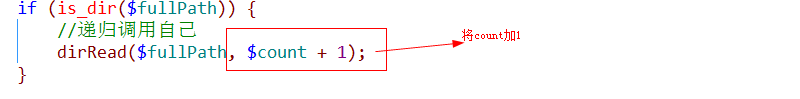
|---

|---

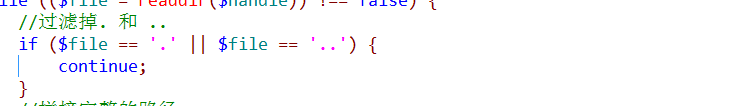
|--

设置一个$count,用于输出前导空格，$count的值需要根据目录的层级，进行改变了，采用递归，只要在每一次递归的时候，将$count的值加上1，这是的count的值依次递增





1. 必须跳出. 和 ..



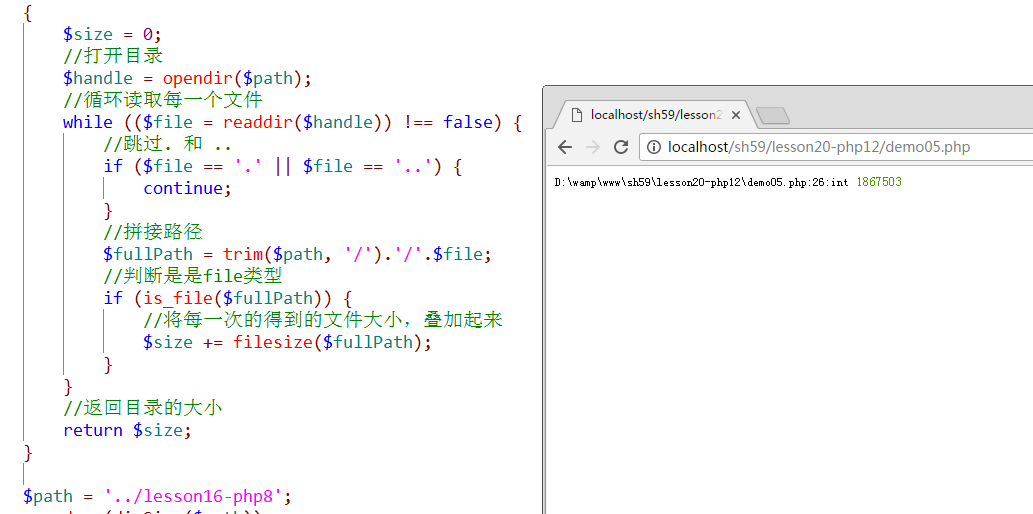
## 递归统计目录的大小

如果是file类型的，直接获取文件的大小

Filesize();

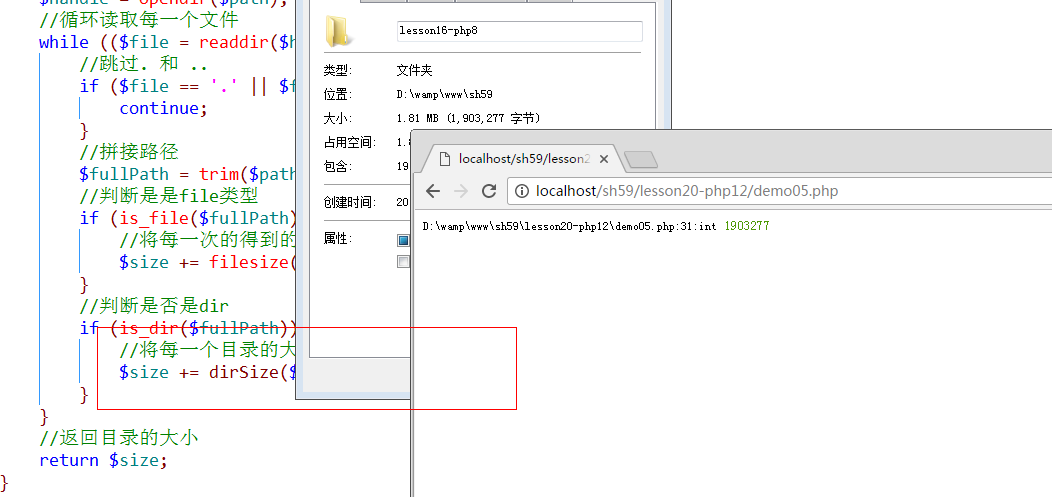
如果是目录需要执行递归函数

1. 直接变量当前的目录下文件的大小



1. 判断里面是否有目录



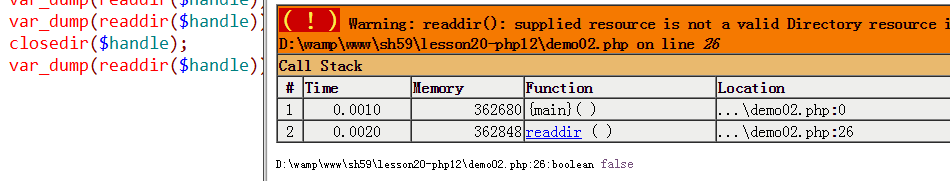
如果是dir执行递归  


没有执行递归的，$size在循环中，是将每一个文件大小叠加在一起。当前函数执行完毕，返回的时候，$size表示的是当前目录的大小，在执行递归的时候，$sise，是每一个目录的大小。

## 关闭目录

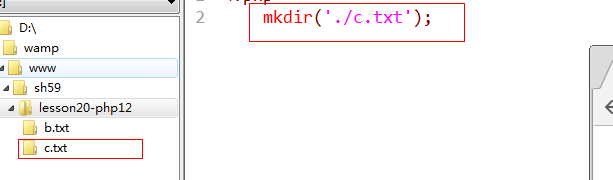
Closedir():销毁打开的目录资源

一旦销毁之后，在不可对该目录进行别的操作了

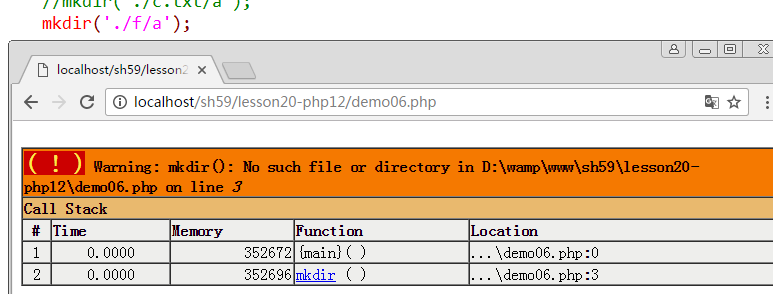


# 创建和删除目录

Mkdir():创建目录



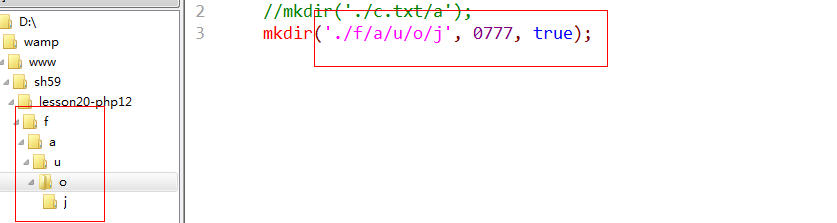
创建目录的时候，必须保证上级的目录的存在



第一个参数：目录路径

第二个参数：目录的权限

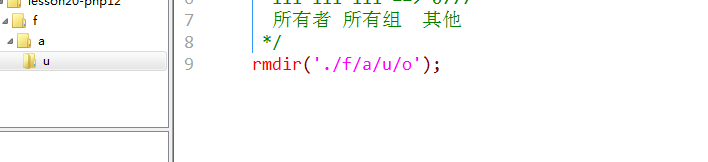
第三个参数：布尔值，true表示执行递归创建



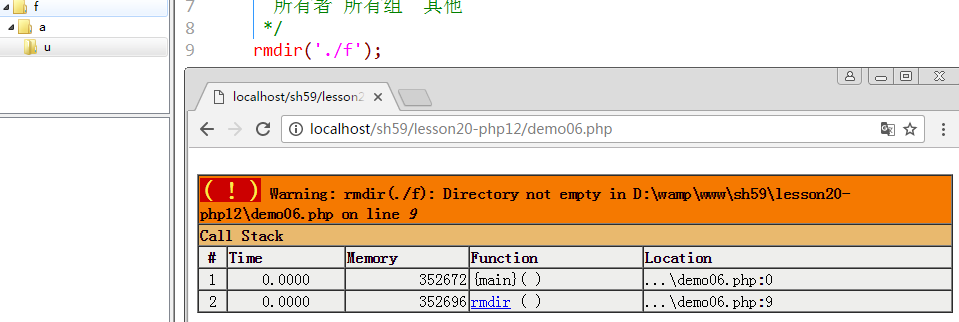
文件的权限



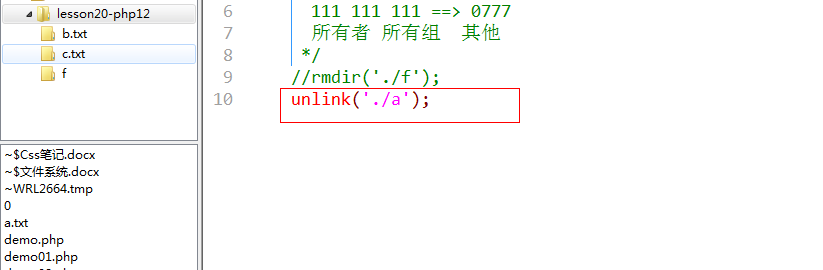
Rmdir():删除目录



一级一级的删除，必须保证目录中的是空的



Unlink():删除文件



不会讲文件放到回收站。

不可以删除目录。

递归删除

目录的复制

文件的操作

文件的上传