

# Ruby Dir 类和方法

**Dir** 是一个表示用于给出操作系统中目录中的文件名的目录流。Dir 类也拥有与目录相关的操作，比如通配符文件名匹配、改变工作目录等。

## 类方法

序号	方法 & 描述
1	<b>Dir[pat]</b> <b>Dir::glob( pat)</b> 返回一个数组，包含与指定的通配符模式 pat 匹配的文件名： <ul style="list-style-type: none"><li>● * - 匹配包含 null 字符串的任意字符串</li><li>● ** - 递归地匹配任意字符串</li><li>● ? - 匹配任意单个字符</li><li>● [...] - 匹配封闭字符中的任意一个</li><li>● {a,b...} - 匹配字符串中的任意一个</li></ul> Dir["foo.*"] # 匹配 "foo.c"、 "foo.rb" 等等 Dir["foo.?"] # 匹配 "foo.c"、 "foo.h" 等等
2	<b>Dir::chdir( path)</b> 改变当前目录。
3	<b>Dir::chroot( path)</b> 改变根目录（只允许超级用户）。并不是在所有的平台上都可用。
4	<b>Dir::delete( path)</b> 删除 path 指定的目录。目录必须是空的。
5	<b>Dir::entries( path)</b> 返回一个数组，包含目录 path 中的文件名。
6	<b>Dir::foreach( path) {   f  ...}</b> 为 path 指定的目录中的每个文件执行一次块。
7	<b>Dir::getwd</b> <b>Dir::pwd</b> 返回当前目录。

8	<b>Dir::mkdir( path[, mode=0777])</b> 创建 path 指定的目录。权限模式可被 File::umask 的值修改，在 Win32 的平台上会被忽略。
9	<b>Dir::new( path)</b> <b>Dir::open( path)</b> <b>Dir::open( path) {  dir  ...}</b> 返回 path 的新目录对象。如果 open 给出一个块，则新目录对象会传到该块，块会在终止前关闭目录对象。
10	<b>Dir::pwd</b> 参见 Dir::getwd。
11	<b>Dir::rmdir( path)</b> <b>Dir::unlink( path)</b> <b>Dir::delete( path)</b> 删除 path 指定的目录。目录必须是空的。

## 实例方法

假设 d 是 Dir 类的一个实例：

序号	方法 & 描述
1	<b>d.close</b> 关闭目录流。
2	<b>d.each {  f  ...}</b> 为 d 中的每一个条目执行一次块。
3	<b>d.pos</b> d.tell 返回 d 中的当前位置。
4	<b>d.pos= offset</b> 设置目录流中的位置。
5	<b>d.pos= pos</b> <b>d.seek(pos)</b> 移动到 d 中的某个位置。pos 必须是一个由 d.pos 返回的值或 0。
6	<b>d.read</b> 返回 d 的下一个条目。
7	<b>d.rewind</b> 移动 d 中的位置到第一个条目。

8	<b>d.seek(pos)</b> 参见 d.pos=pos。
9	<b>d.tell</b> 参见 d.pos。

 点我分享笔记