

## python 的 os 和 shutil 模块对应的 shell 命令

20160829

python	shell	描述
os.getcwd()	pwd	当然文件夹绝对路径，一样
os.getenv("env")	echo \$env	获取环境变量，一样
os.listdir("path")	ls path	只相当于不带参数的 ls
os.remove("filepath")	rm filepath	相当于不带参数的 rm，不能删除文件夹
os.rmdir("dirpath")	rmdir dirpath rm -r dirpath	删除文件夹
os.system("cmd")		
os.mkdir("dirpath")	mkdir dirpath	
os.makedirs("ab/cd/ef")	mkdir ab mkdir ab/cd mkdir ab/cd/ef	递归创建文件夹 os.makedirs("a")==os.mkdir("a")
os.chdir("dirpath")	cd "dirpath"	改变工作目录，ShellCmd("cd dirpath")不起作用
os.rename("path1", "path2")	mv path1 path2	功能一样，移动重命名
os.path.split("a/b/c") => ["a/b", "c"] os.path.split("a/b/c/") => ["a/b/c", ""]		将一个路径拆分成“文件夹名”(末尾无"/")和“文件名”
os.path.dirname("a/b/c") => "a/b" os.path.dirname("a/b/c/") => "a/b/c"		返回文件夹名，末尾没有"/"
os.path.basename("a/b/c") => "c" os.path.dirname("a/b/c/") => ""		返回文件名
os.path.isfile("path")		判断一个路径是 file 或是 directory

os.path.isdir("path")		
os.path.exists("path")		判断路径是否存在, 对于文件夹路径, 末尾有没"/"结果一样
os.path.getsize("path")		获得文件的大小, 单位 Byte
os.path.expanduser("path")		将 path 中的 "~" 用绝对路径来代替
os.path.abspath("path")		返回绝对路径, 但注意, 并不展开 "~": os.path.abspath("~/a") => "/Users/huang/study/~a"
os.path.splitext("path") os.path.splitext("a/b.txt") => ["a/b", ".txt"] os.path.splitext("a/b.c/d") => ["a/b.c/d", ""]		拆分文件名和扩展名
os.path.join("a/", "b", "c/", ...) => "a/b/c"		先去掉 a,b,c,... 末尾的 "/", 再用 "/" 按顺序将 a,b,c,... 连接起来, 末尾没有 "/"
shutil.copyfile("a/b/c", "x/y/c")	cp a/b/c x/y/c	复制普通文件。注意, shell 中可以不指定复制后的文件夹, 但 python 中要指定
shutil.copy("a/b/c", "x/y/c") shutil.copy("a/b/c", "x/y/") shutil.copy("a/b/c", "x/y")	cp a/b/c x/y/c cp a/b/c x/y/ cp a/b/c x/y	两者完全一样, 比 shutil.copyfile() 好用
shutil.copy2("a/b/c", "x/y/")	cp -p a/b/c x/y/	即复制文件内容(shutil.copy), 也复制最后访问时间和修改时间
shutil.copytree("dir1", "dir2", True/False)	cp -r dir1 dir2	递归复制文件夹。如果第 3 个参数是 True, 则复制目录时将保持文件夹下的符号连接, 如果第 3 个参数是 False, 则在复制的目录下生成物理副本来替代符号连接
shutil.move(a, b) == os.rename(a, b)	mv a b	功能一样
shutil.rmtree("dirpath")	rm -r dirpath	dirpath 只能是文件夹 directory