



OS

1. readme

os库用处较多，在任何开源项目里都能看到这个库的身影，比如在python中查看当前py文件的路径，或者是移动将某个文件移动至另一个文件夹。

os全称operating system，提供各种python程序与操作系统的接口。本次文档使用代码中python文件的绝对路径是：`'/Users/blk/Desktop/test.ipynb'`，注意路径中是 `/` 还是 `\`，mac是 `/`，linux是 `\`。建议自己找一个文件夹和文件手动运行一次保留印象，之后只需看左边的目录快速查找即可。

2. 相对路径与绝对路径

- 相对路径是相对于当前工作目录或者其他已知位置的路径。它描述了如何到达一个文件或目录，而不需要指定完整的路径。相对路径通常使用当前目录（`.`）和父目录（`..`）来表示路径。例如，在当前目录下的子目录 `docs` 中有一个文件 `example.txt`，那么相对于当前目录的路径就是 `./docs/example.txt`。
- 通俗理解来看 `'/Users/blk/Desktop/test.ipynb'` 可以表示为 `'./test.ipynb'`，那么此时要表示Desktop路径下的文件都可以用 `./文件名` 来表示。
- 表示blk路径下的文件都可以用 `../文件名` 来表示。因为 `blk` 算是 `Desktop` 的父目录。
- linux等系统的绝对路径一定以 `/` 开头

3. os

3.1 获取该文件所在的文件os.getcwd()

```
1 os.getcwd()
```

- output: '/Users/blk/Desktop'

3.2 更改当前工作目录os.chdir(path)

```
1 # os.chdir(path)
2
3 os.chdir('/Users/blk/Desktop')
```

- 函数用途：在任何一个文件夹里面运行程序都可以把当前的目录变成 Desktop，然后运行该文件夹 Desktop 里面的文件

3.3 查看当前环境变量

```
1 os.environ
```

- 这个函数之后在配置openAI的api时会使用较多

3.4 获取父目录 os.pardir

```
1 print(os.pardir)
```

- Output: ..

4. os.path

4.1 获取文件中的文件名os.path.basename(path)

```
1 # os.path.basename(path)
2 # 返回为绝对路径
3
4 os.path.basename('test.ipynb')
```

- output: 'test.ipynb'
- 注：个人觉得这个函数没什么用

4.2 获取文件中的绝对路径os.path.abspath(path)

```
1 # os.path.abspath(path)
2 # 返回一个绝对路径
3
4 os.path.abspath('test.ipynb')
```

- output: '/Users/blk/Desktop/test.ipynb'

4.3 将多个路径组合返回os.path.join(path1,path2,...)

```
1 # os.path.join(path1,path2,...)
2 # 返回为path1/path2
3
4 abspath = os.getcwd()
5 text_file = 'test.ipynb'
6 os.path.join(abspath,text_file)
7
8 # 效果等同于 abspath+'test.ipynb'
9 # 效果等同于 os.path.abspath('test.ipynb')
```

- output: '/Users/blk/Desktop/test.ipynb'

4.4 返回文件路径的目录os.path.dirname(path)

```
1 # os.path.dirname(path)
2 # 返回指定路径的目录名，不包括文件名
3
4 os.path.dirname('/Users/blk/Desktop/test.ipynb')
5 # 返回 '/Users/blk/Desktop'
6
7 os.path.dirname('blk/Desktop/test.ipynb')
8 # 返回 '/blk/Desktop'
```

- 这个比较常用

4.5 判断路径是否存在os.path.exists/isfile/isdir(path)

```
1 # 判断路径是否存在
2 os.path.exists(path)
3
4 # 判断目录是否存在
5 os.path.isdir(path)
6
7 # 判断文件是否存在
8 os.path.isfile(path)
```

4.6 列出目录下的所有文件和子目录os.listdir(path = '.')

```
1 os.listdir(path = '.')
```

4.7 创建一个目录os.mkdir(path)

```
1 os.mkdir('文件夹0')
```

- 可以简单理解为创建一个文件夹

4.8 递归创建文件夹os.makedirs(path,exist_ok = False)

```
1 os.makedirs('文件夹1/文件夹2/文件夹3')
2
3 # exist_ok选项默认为False
4 # 当设置为True时若父文件夹存在则会报错
5 # 比如上文已经创建了一个名为'文件夹0'当文件夹，再执行
6 os.makedirs('文件夹/文件夹2/文件夹3')
7 #会报错文件'文件夹0'已经存在
```

4.9 删除一个文件夹os.remove(path)

```
1 os.remove('test.ipynb')
```

4.10 重命名文件或文件夹os.rename(path)

```
1 os.rename('test.ipynb','new_name.ipynb')
```

- 把文件 test.ipynb 改名为 new_name.ipynb
- 方法同样用于更改文件夹的名字

4.11 列出目录下的所有文件os.listdir(path)

- 假设path目录下有三个文件 `file1.txt` , `file2.txt` , `file3.txt`

```
1 os.listdir(path)
```

• output: [file1.txt, file2.txt, file3.txt]