

test

2025 年 4 月 22 日

0.1 About 1

0.2 About 2

0.3 关于

```
[5]: print("Hello Python")
      print("Again")
      print("Third")
```

Hello Python

Again

Third

0.4 路径操作

功能	方法/属性	示例
获取当前目录	<code>Path.cwd()</code>	<code>Path.cwd()</code>
获取用户主目录	<code>Path.home()</code>	<code>Path.home()</code>
路径拼接	<code>/</code> 运算符	<code>Path("dir") / "file.txt"</code>
转换为字符串	<code>str(path)</code>	<code>str(Path("file.txt"))</code>

0.5 路径信息

功能	方法/属性	示例
检查路径是否存在	<code>path.exists()</code>	<code>Path("file.txt").exists()</code>
判断是否为文件	<code>path.is_file()</code>	<code>Path("file.txt").is_file()</code>
判断是否为目录	<code>path.is_dir()</code>	<code>Path("dir").is_dir()</code>

功能	方法/属性	示例
获取文件名	<code>path.name</code>	<code>Path("dir/file.txt").name</code>
获取文件名（无后缀）	<code>path.stem</code>	<code>Path("file.txt").stem</code>
获取扩展名	<code>path.suffix</code>	<code>Path("file.txt").suffix</code>
获取父目录	<code>path.parent</code>	<code>Path("dir/file.txt").parent</code>

0.6 文件/目录操作

功能	方法	示例
遍历目录	<code>path.iterdir()</code>	<code>for f in Path("dir").iterdir()</code>
查找文件（非递归）	<code>path.glob("*.txt")</code>	<code>Path("dir").glob("*.txt")</code>
查找文件（递归）	<code>path.rglob("*.txt")</code>	<code>Path("dir").rglob("*.txt")</code>
读取文件内容	<code>path.read_text()</code>	<code>Path("file.txt").read_text()</code>
写入文件内容	<code>path.write_text()</code>	<code>Path("file.txt").write_text("Hello")</code>
创建目录	<code>path.mkdir()</code>	<code>Path("new_dir").mkdir()</code>
创建多级目录	<code>path.mkdir(parents=True)</code>	<code>Path("a/b/c").mkdir(parents=True)</code>
删除文件	<code>path.unlink()</code>	<code>Path("file.txt").unlink()</code>
删除空目录	<code>path.rmdir()</code>	<code>Path("empty_dir").rmdir()</code>

0.7 高级功能

功能	方法	示例
获取绝对路径	<code>path.resolve()</code>	<code>Path("file.txt").resolve()</code>
打开文件	<code>path.open()</code>	<code>with Path("file.txt").open() as f:</code>
获取文件状态	<code>path.stat()</code>	<code>Path("file.txt").stat()</code>