

Files

Python Reference

Reading & writing text files

A text file 文本文件 stores text on disk. Open it with `open(name, mode)` where mode 模式 says read or write. Always use `with`, which closes the file for you.

Writing

Mode "w" writes a new file and erases any old one.

```
with open("notes.txt", "w") as f:
    f.write("first line\n")
    f.write("second line\n")
print("saved")           # saved
```

Reading

Mode "r" (the default) reads. `.read()` returns the whole file as one string.

```
with open("notes.txt", "w") as f:
    f.write("hello\nworld\n")
with open("notes.txt") as f:
    print(f.read().strip())    # hello / world
```

Line by line

Loop over the file to get one line at a time. `.strip()` removes 去除 the newline 换行符 at the end.

```
with open("data.txt", "w") as f:
    f.write("Mei,88\nSam,71\n")
with open("data.txt") as f:
    for line in f:
        name, score = line.strip().split(",")
        print(name, "scored", score)
# Mei scored 88
# Sam scored 71
```

Appending

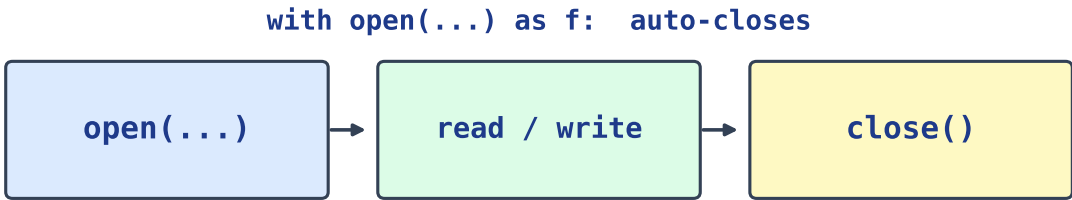
Mode "a" appends 追加 — it adds to the end without erasing.

```
with open("log.txt", "w") as f:
    f.write("line 1\n")
with open("log.txt", "a") as f:
    f.write("line 2\n")
with open("log.txt") as f:
    print(f.read().strip())    # line 1 / line 2
```

Mode	Meaning
"r"	read (default)
"w"	write (erases first)
"a"	append (add to the end)

Common mistakes

- Always close a file, or use `with open(...) as f:` which closes it for you.
- `read()` gives the whole file as one string, and each line still ends with `\n`.
- Opening with "w" erases the file first; use "a" to add to the end.



Exam tip: mode 'r' reads, 'w' writes (overwrites); prefer with so the file always closes

open → read/write → close (with auto-closes)