

File Input & Output

[References](#)

[File input](#)

[File output](#)

[Appending](#)

[Multiple files](#)

[Safe File I/O](#)

[Python3 'with' keyword](#)

References

- tutorialspoint: [Python Files I/O](#)
- wikibooks: [Python Programming/Input and Output](#)

File input

- Use **open('filename', 'r')** to open a file in '**read**' mode
- The input file should be located in the same directory as the script.

```
infile = open('infile.txt', 'r')
for line in infile:
    #line.strip() will remove extra whitespace & line breaks
    line = line.strip()
    print( line)
infile.close()
```

- You can also use the **readline()** function

```
infile = open('infile.txt', 'r')
line = "line"
while(line):
    line = infile.readline().strip()
    print(line)
```

Tokenizing

- Splitting a line up into words is called "tokenizing,"
- It's common to split by whitespace, or by comma (e.g. CSVs) or semicolons

```
line = infile.readline().strip()
tokens = line.split() # splits using whitespace by default
```

File output

- Use `open('filename', 'w')` to open a file in **'write'** mode
 - If the outfile doesn't exist, it will be created in the same directory as the script.
 - If the file exists, its contents will be cleared / overwritten.
- The examples below just write one line to a file.
 - Note that you must use `"\n"` to write a line break.

```
outfile = open("outfile.txt", "w")
outfile.write("Hello file!\n")
outfile.write("line 2\n")
outfile.close()
```

Appending

- Use 'a' mode to append to a file

Multiple files

- You can have more than one file open at a time:
- This code copies the contents of `infile.txt` to `outfile.txt`.

```
infile = open('infile.txt', 'r')
outfile = open("outfile.txt", "w")
for line in infile:
    outfile.write(line)
infile.close()
outfile.close()
```

Safe File I/O

- Use `try/except`
- Finally will ensure that the file is closed

```
def safeFileIO():
    f = None
    try:
        f = open("asdf.txt")

        for line in f:
            print(line)
    except:
        print("something bad happened")
    finally:
        f.close()

safeFileIO()
```

Python3 'with' keyword

- Input:

```
with open('infile.txt', 'r') as infile:
    for line in infile:
        print( line, end="" )
```

- Output

```
with open("outfile.txt", "w") as outfile:
    outfile.write("Hello file!\n")
    outfile.write("line 2\n")
```

- Multiple files

- You can open multiple files using a single **with** statement
- This example copies the contents of infile.txt to outfile.txt

```
with open('infile.txt', 'r') as infile, open("outfile.txt", "w") as outfile:
    for line in infile:
        outfile.write(line)
```

- Safe I/O

- Use a **with** statement
- This way is cleaner to ensure safe I/O

```
def safeFileIO():
    try:
        with open("asdf.txt") as f:
            for line in f:
                print(line)
    except:
        print("something went wrong")

safeFileIO()
```