
Part V

File I/O

File I/O

Opening files

- Create file object (text files)

```
fd = open("testfile.txt")           # read
fd = open("testfile.txt", 'r')      # read
fd = open("testfile.txt", 'w')      # write
fd = open("testfile.txt", 'a')      # append
```

- Create file object (binary files)

```
fd = open("testfile.txt", 'rb')     # read
fd = open("testfile.txt", 'wb')     # write
fd = open("testfile.txt", 'ab')     # append
```

- `open` has more options (encoding, how to handle newlines, ...)

File I/O (2)

Methods of file objects

- Reading

```
fd.read()           # all
fd.read(n)          # n Bytes
fd.readline()       # one line
fd.readlines()      #
```

- Writing

```
fd.write("New_Text\n")
fd.writelines(["first_line\n", "Second\n"])
```

- Don't forget to write newlines
- Closing

```
fd.close()
```

File I/O (3)

Iterating over textfile

- `for` loop over file line by line

```
fd = open("fulltext.txt")
for line in fd:      # rather than: in fd.readlines()
    print(line)
# or:
while True:
    line = fd.readline()
    if not line:
        break
    print(line)
```

Some more functions on file objects

- `fd.tell()` – get current file position
- `fd.seek(offset)` – set file to position
- `fd.flush()` – flush output buffer. Note: buffered for efficiency

Module *pickle*

Save Python objects directly to file

```
import pickle
fd = open('myObjects', 'wb')
pickle.dump(x, fd)
pickle.dump(y, fd)
fd.close()
```

- Most objects can be pickled
- Then load in same order

```
fd = open('myObjects', 'rb')
x=pickle.load(fd)
y=pickle.load(fd)
fd.close()
```