1811/2807/7001ICT
Programming Principles

School of Information and Communication Technology
Griffith University

Trimester 1, 2024

20 Files

As a scripting language, one of tasks Python is applied to very often is processing text files.

20.1 File handles

All of the information about a file you are working with in a Python program is stored in an object.

A variable that refers to one of these objects is called a *file handle*.

20.1.1 Predefined handles

Module `sys` defines three file handles:

- `sys.stdin` – the standard input stream, usually for user input via the keyboard;
- `sys.stdout` – the standard output stream, usually for output to the console; and
- `sys.stderr` – the standard error stream, usually for output of error messages to the console.

You don't need to open these "files", or close them.

20.1.2 Opening files

The built-in function `open` is used to open a file so your program can work with it.

It has more optional, keyworded arguments, but this definition is usually sufficient:

`open(`*`path`*`, mode = 'r')`

returns a file handle, for the file at *`path`*, using the specified mode, where the commonly used modes are:

`'r'`	read text (default)
`'w'`	write (clearing the file first)
`'a'`	append (write to the end of the existing file)

20.1.3 Closing files

After opening, a file is operated on with the methods of file object.

If you open a file, it is your responsibility to close it when you are done with it.

Use the `close` method.

20.1.4 Reading and writing

Most commonly, text files are read a line at a time.

Use the `readline()` method.

This *only* returns an empty string if the end of the file has been reached.

A blank line will contain at least the terminating newline.

To write text to a file, use the `write()` method.

This example program copies the text from one file to another.

```
# file: file1.py
# Ask the user for two file names. Copy the contents of
# the first file to the second, using readline().
```

```
path1 = input("What file to copy from: ")
f1 = open(path1)
path2 = input("What file to copy to: ")
f2 = open(path2, mode = 'w')
line = f1.readline()
while line != "":
   f2.write(line)
   line = f1.readline()
f1.close()
f2.close()
```

20.1.5 Reading with a for loop

An input file object may be used as if it is a sequence of lines in a `for` loop.

```
# file: file2.py
# Ask the user for two file names. Copy the contents of
# the first file to the second, using a for loop.

f1 = open(input("What file to copy from: "))
f2 = open(input("What file to copy to: "), mode = 'w')
for line in f1:
   f2.write(line)
f1.close()
f2.close()
```

20.2 Text processing

With all the `str` methods Python is a formidable tool for processing data saved in text files, and that is one of its most frequent uses.

20.2.1 Text processing example problem

Complete this problems as a class.

> *Problem:* Prompt for the name of a Python script. Read it and print the names of all of the modules it imports from. For example, if `fred.py` contains:
>
> ```
> # file: fred.py
>
> import judy
> from gert import harry
> import frank
>
> blah
> blah
> blah
> ```

The program should look like this:

```
Enter script name: fred.py
Modules imported: judy, gert, frank.
```

Hint: use `str.split`.

Section summary

This section covered:

- file handles;
- opening and closing files;
- reading from and writing to files; and
- text processing.