



Files

ASLAM SAFLA
<ASLAM@CS.UCT.AC.ZA>



Storing data for the future

Data in a computer is stored in files

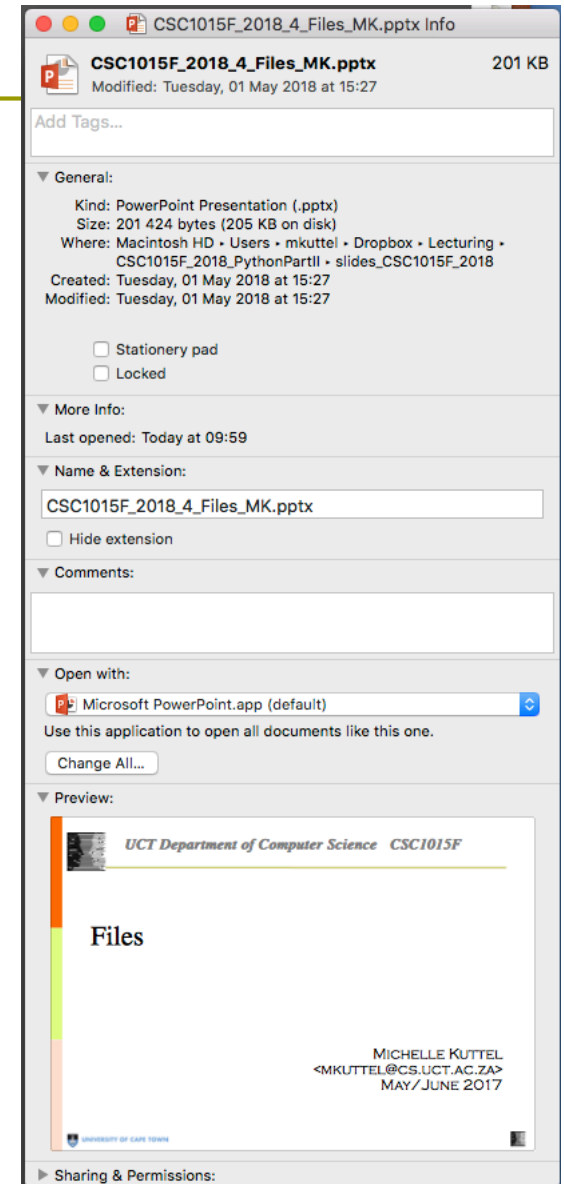
- ▣ You will have used these before
 - ▣ Word has .doc files
- ▣ If you want your computer to use data collected before, or by others, you will need to interact with files.
 - ▣ Especially if you have a lot of data.



Concept: Files

An file is a sequence of **bytes** stored on a secondary storage device e.g., hard drive.

- Files are usually stored in **folders** (or **directories**).
- Every file has a filename
 - the rules for valid filenames are determined by the operating system (OS).
 - Additional information about each file (e.g., last modification date) is managed by the OS.



Problem

- Write a program to count the number of lines in a text file.



Concept: File Types

- ❑ **Text files** are made up of strings, each of which is terminated with a **newline**.
 - The file can be processed one line at a time.
- ❑ **Binary files**, which are not text files, store information in application-specific formats e.g., PDF, JPEG.
 - ❑ Compiled programs are stored as binary files

All text files are binary files, but not vice versa!



Opening and Closing Files

Files must be **opened** before their contents can be read.

Syntax:

```
<file> = open (<filename>, <mode>)
```

filename is the name of the file.

mode is one of: “r” to read; “w” to overwrite; “a” to append

Example:

```
f = open ("data.txt", "r")
```

Files must be closed when you are done.

```
f.close ()
```



Options for read functions

`f.read([n])` Reads at most `n` bytes.

`f.readline([n])` Reads a single line of input up to `n` characters. If `n` is omitted, this method reads the entire line.

`f.readlines([size])` Reads all the lines and returns a list. `size` optionally specifies the approximate number of characters to read on the file before stopping.



Reading Text Data

Multiple ways to do this in Python.

- ▣ Read **entire file** into **one string**.

```
f.read()
```

- ▣ Read **next line** of text into **a string**.

```
f.readline()
```

- ▣ Read **entire file** into a **list of strings**.

```
f.readlines()
```

- ▣ **Iterate over file** and read in one line at a time.

```
for line in f:  
    do_something (line)
```



Problem

- Write a program to count the number of lines in a text file.

More: Opening and Closing Files

There are a number of options you can give to `open()` (see `open.__doc__`)

encoding is the name of the encoding used to decode or encode the file. This should only be used in text mode. The default encoding is platform dependent, but any encoding supported by Python can be passed.

‘encoding’ is a useful one for files that contain non-ascii characters (default encoding is **ascii**). E.g. :

```
myFile=open(filename, 'r', encoding='utf-8')
```

UTF-8 is a character encoding capable of encoding Unicode characters.



Problem

- Write a program to count the frequency of each word in a text file.



Problem

Write a program to search a text file like a simple (albeit inefficient) database.

- A database stores information that can be manipulated/searched efficiently.

Each line should contain fields separated by commas, and the program should allow searching of any field.

For example:

- Mortimer,Mouse,22
- Spongebob,Squarepants,15
- James,Sullivan,37



Writing Text Data

Write out text data using a **print** statement with additional *file* attribute.

□ Syntax:

```
print (<expr1>, <expr2>, ..., file=<file>)
```

□ Example:

```
print ("Hello World", file=f)
```

`f.write(s)` Writes string `s`.

`f.writelines(lines)` Writes all strings in sequence `lines`. Can deal with a list of strings. Useful if want to add one file to another.



Problem

Write a program to store a list of data observations to a text file.

▣ For robustness, “save” the file after every entry.



Problem

- Write a Python program to write Python programs.
 - Put these programs in a separate folder/directory

Recursion!

Write a recursive function `hat` that behaves as follows:

```
>>>hat("hamster")
```

```
s
```

```
mst
```

```
amste
```

```
hamster
```

```
>>>hat("california")
```

```
fo
```

```
ifor
```

```
liforn
```

```
aliforni
```

```
california
```



Problem

Write a program to print out the first 3 lines of a file.

- ▣ Deal with possible file IO errors with exception handling.



Exceptions

An exception is a runtime error that can be checked for and dealt with in a program.

□ Syntax:

```
try:
    <statements>
except <Exception1Name>:
    <statements>
except <Exception2Name> as <parameter>:
    <statements>
except:
    <statements>
finally:
    <statements>
```



Exception Semantics

Python will try to execute the code in **try**.

- ❑ If an error occurs, the **except** clauses will be searched in sequence.
 - If there is a match, the code in the except is executed.
 - If no match, the exception is propagated to the calling function.
- ❑ After the try code and/or except code executes, the **finally** code is executed, typically to do cleanup such as close files.



Exception Example

```
try:
    f = open ( "thetest.txt", "r" )
    data = f.read()
except IOError as errno:
    print ( "Could not read file" )
    print ( "Error number:",errno)
finally:
    print ( "cleaning up" )
```



Common Exceptions

- ❑ IOError error number
- ❑ NameError
- ❑ TypeError
- ❑ ValueError
- ❑ ZeroDivisionError
- ❑ IndexError
- ❑ ...
- ❑ [see <http://docs.python.org/py3k/library/exceptions.html>]



Improve “Hello Country”

With exception handling

Finding out whether a file exists

- Catch an IO error
- Import os.path

```
import os.path
if os.path.isfile("myData.txt"):
    print("File exists!")
else: print("File does not exist!")
```



Questions from class

- When dealing with file is it possible to access in more than one mode i.e. appending and reading at the same time?
 - The file mode is '**r**' for read, '**w**' for write, or '**a**' for append. A file can be opened for in-place updates by supplying a plus (+) character, such as '**r+**' or '**w+**'. Be careful with this.
 - A mode of '**w+**' truncates the file to zero length if it already exists. A mode of '**r+**' or '**a+**' opens the file for both reading and writing but leaves the original contents intact when the file is opened.

