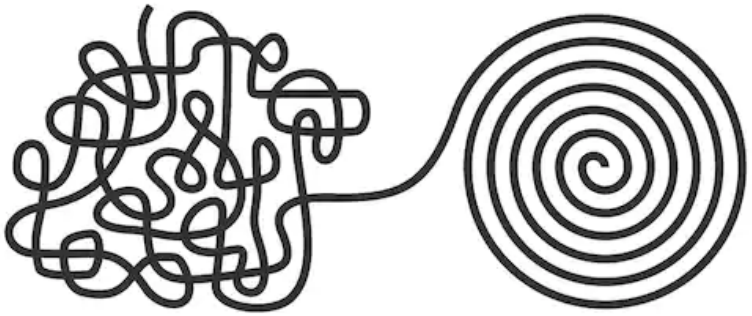
File Handling

COMP16321 – Introduction to Programming 1

Gareth Henshall
Lecturer in Computer Science

Dictionaries Vs. Lists

Dictionaries	Lists
Unordered	Ordered
Fetched by Key	Fetched by Position



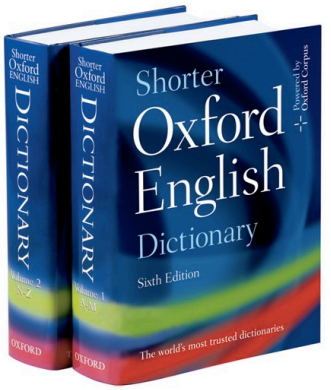
Python Dictionaries

Accessed by a Key not Position

Unordered collection of arbitrary objects

Variable-length, heterogeneous and arbitrarily nestable

Of the category “mutable mapping”- cannot rely on index



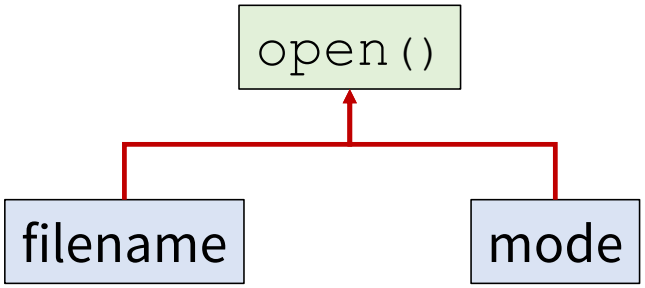
Common Dictionary Literals and Operations

`D = {}`	Empty Dictionary
`D = {'spam': 2, 'eggs': 3}`	Two Item Dictionary
`D = {'food':{'ham': 1, 'cheese': 2}}`	Nested Dictionaries
`D = dict(name = 'Bob', age = 40)`	Alternative Construct
`D.keys()`	List Of All Keys
`D.values()`	List Of All Values
`D['bread'] = 3`	Add an Entry to the Dictionary
`D.get('ham')`	Returns the Value Associated to ham
`D.clear()`	Removes All Items

Python File Handling

General Operations:

- Create
- Read
- Write
- Delete



Opening a File

There are 4 different methods (modes) for opening a file:

"r" (Read):

- Default value.
- Opens a file for reading.
- Error if the file does not exist

"a" (Append):

- Opens a file for appending.
- Creates a new file if it doesn't exist

"w" (Write):

- Opens a file for writing.
- Creates the file if it doesn't exist

"x" (Create):

- Creates the specified file
- Returns an error if the file exists

Opening a File Continued

You can specify if the file should be handled in Text or Binary mode:

`"t"` (Text):

- Default Value
- Text Mode

`"b"` (Binary):

- Binary Mode (e.g. images)

Basic Syntax

To open a file it is enough to just specify the name of the file:

```
file = open("demoFile.txt")
```

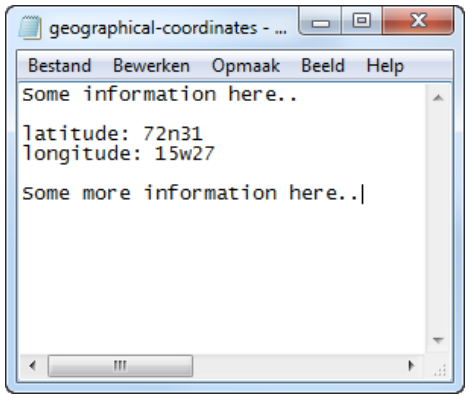
Both of the code samples are equivalent because "r" & "t" are default values.

```
file = open("demoFile.txt", "rt")
```

Reading a File

```
file = open("demoFile.txt", "r")
print(file.read())

file.close()
```



Writing to a File

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()
```

Adds the text to the bottom of the file

Overwrites the text in the file

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()
```

Deleting a File

To remove a file you need to use the OS module.

```
import os
os.remove("demofile.txt")
```