



COMP10001

Foundations of Computing

Semester 1, 2021

Tutorial 9

Andrew Naughton

andrew.naughton@unimelb.edu.au

Outline

- ❖ Libraries
 - ❖ `defaultdict`
 - ❖ `list` comprehensions
- ❖ Iterators
 - ❖ `itertools`
- ❖ Files & Operations
- ❖ CSVs
- ❖ Exercises

Libraries

- ❖ A library contains a group of resources (e.g. functions, methods, and variables) that extend Python to perform more diverse operations
- ❖ They can save you time as you don't have to implement those functions/methods/variables by yourself

- ❖ Import entire library:

```
import < library/ module >
```

```
from < library/ module > import *
```

- ❖ Import specific functions:

```
from < library/ module > import < function name >
```

defaultdict

- ❖ Works the same way as a normal `dict`, except that:
 - ❖ The keys are initialised with a default value
 - ❖ Will not raise a `KeyError` if the key doesn't exist – instead, it creates a new key with a default value and returns that default value

```
from collections import defaultdict as dd

def get_char_frequencies(text):

    char_frequencies = dd(int)
    for char in text:
        char_frequencies[char] += 1

    return char_frequencies
```

list comprehensions

- ❖ Shortcut notation
- ❖ Allows us to create a list (that would otherwise require a loop) using one expression
- ❖ Format:

```
new_list = [< expr > for var in < iterable > < optional condition >]
```

list comprehensions

- ❖ Can be used for:
 - ❖ Mapping
 - ❖ *“Square all numbers in a list”*

```
nums_squared = [num ** 2 for num in nums]
```

- ❖ *“Map all numbers to either odd or even”*

```
odd_or_even = ["odd" if num % 2 else "even" for num in nums]
```

- ❖ Filtering
 - ❖ *“Extract all even numbers”*

```
nums_even = [num for num in nums if num % 2 == 0]
```

Iterators

- ❖ Essentially an object which is used to iterate over other objects (iterables: list, tuple, set, dict, str)
- ❖ Functions used for iterators:
 - ❖ Constructs an iterator out of an **iterable**
 - ❖ Returns the new element in the **iterator**; or throws error if reached the end
- ❖ Useful as we don't need to store the entire container object in **memory** to iterate over it – only the iterator object

```
iter(<iterable>)
```

```
next(<iterator>)
```

itertools

- ❖ A **library** which provides many methods to construct iterators
- ❖ Produces an iterator to cycle through an iterable, **looping** from the end back to the beginning **infinitely**
`cycle(iterable)`
- ❖ Produces an iterator containing tuple elements, created by taking the **cartesian product** of the iterables specified
`product(iterables[, repeat])`
- ❖ Produces an iterator of every possible combination of elements from the iterable
`combinations(iterable, group_size)`
- ❖ Also: `permutations(iterable, group_size)` `groupby(iterable[, keyfunc])`

Files

- ❖ Allow us to store data on a computer permanently
- ❖ Will persist on storage media after a program is terminated
- ❖ Are also useful for storing large amounts of data in a structured way and sharing it with others

File Operations

- ❖ Opening a file: `file = open(path_to_file, file_mode)`
- ❖ File modes:
 - 'r' File will only be read (from the start) [default value]
 - 'w' File will only be written to
If the file already has data, that data will be deleted
 - 'a' Same as 'w' except will append to any existing data

File Operations

- ❖ Reads entire file and returns its contents as **str**: `.read()`
- ❖ Reads a single line from file and returns it as **str**: `.readline()`
- ❖ Reads entire file and returns **list** where each element corresponds to a line in file: `.readlines()`
- ❖ Takes a **str** and writes that to end of file `.write(line: str)`
- ❖ Takes a **list** of **strs** and iteratively writes them to end of file `.writelines(lines: list)`

File Operations

- ❖ Closing a file: `file.close()`
- ❖ You should always close files when you're done with them

CSVs

- ❖ A **CSV** (Comma Separated Values) file is a text file where the information is organised in a format similar to that of a **spreadsheet**
- ❖ Each line (**row**) represents a record
 - ❖ Separated by `\n`
 - ❖ The first row is often a header with the names of the different fields
- ❖ Each **column** represents a record's value for a field
 - ❖ Separated by `,`
- ❖ Useful for storing information like **statistics** or **measurement** data

```
Name,Position,Nickname
Dewey Finn,Lead Singer/ Guitarist,Mr.S
Summer Hathaway,Band Manager,Tinkerbelle
Zack Mooneyham,Lead Guitarist,Zack-Attack
```



Exercises