

INF 502 – SOFTWARE DEVELOPMENT METHODOLOGIES

Python: advanced concepts

Dealing with Files

A file in Python serves as a link to an actual file from the computer

| | |
|--|---------------------------------------|
| <code>file_handle = open('data.csv', 'r')</code> | Open the csv file in read mode |
| <code>content = file_handle.read()</code> | Read whole file content into a string |
| <code>content = file_handle.read(N)</code> | Reads N bytes (N >= 1) into a string |
| <code>line = file_handle.readline ()</code> | Read one line from the file |
| <code>lines = file_handle.readlines()</code> | Returns a list of line strings |
| <code>file_handle.close()</code> | Close the file |

Reading from a file

```
file_handler = open('lines.csv', 'r')
for line in file_handler.readlines():
    print (line)
file_handler.close()
```

['Line 1, animals, 1, 0\n', 'Line 2, vegetables, 0, 5\n', 'Line 3, animals, 2, 3\n', 'Line 4, minerals, 1, 1\n', 'Line 5, vegetables, 2, 2\n']

The diagram illustrates the output of the provided Python code. A list of five CSV-formatted lines is shown on the left. Two red circles highlight the first two elements of the list. Blue lines with arrows originate from these circles and point to a larger box on the right. This box contains the same five lines, but they are formatted as plain text, with the trailing '\n' characters removed. This represents the state of the file after the first two lines have been read and printed.

Line 1, animals, 1, 0
Line 2, vegetables, 0, 5
Line 3, animals, 2, 3
Line 4, minerals, 1, 1
Line 5, vegetables, 2, 2

Reading from a file

```
file_handler = open('lines.csv', 'r')
for line in file_handler.readlines():
    print (line.rstrip('\n'))
file_handler.close()
```

```
['Line 1, animals, 1, 0\n', 'Line 2, vegetables, 0, 5\n', 'Line 3, animals, 2, 3\n', 'Line 4, minerals, 1, 1\n', 'Line 5, vegetables, 2, 2\n']
```

```
[...]
Line 1, animals, 1, 0
Line 2, vegetables, 0, 5
Line 3, animals, 2, 3
Line 4, minerals, 1, 1
Line_5, vegetables, 2, 2
```

Reading from a file

```
file_handler = open('lines.csv', 'r')
for line in file_handler.readlines():
    line = line.rstrip('\n')
    field = line.split(',') #break the string into a list (split - comma)
    print (field)
file_handler.close()
```

```
['Line 1', ' animals', ' 1', ' 0']
['Line 2', ' vegetables', ' 0', ' 5']
['Line 3', ' animals', ' 2', ' 3']
['Line 4', ' minerals', ' 1', ' 1']
['Line 5', ' vegetables', ' 2', ' 2']
```

Writing to a file

| | |
|--|---|
| <code>file_handle = open('data.csv', 'w')</code> | Open the file 'data.csv' in write mode |
| <code>content = file_handle.write(S)</code> | Write string S to a file |
| <code>content = file_handle.writelines(L)</code> | Write the strings in the list L to a file |
| <code>file_handle.close()</code> | Close the file |

Writing to a file

```
file_handler = open('other.csv', 'w')  
list_1 = ('banana', 'carrot', 'avocado', 'orange', 'grapefruit')  
file_handler.writelines(list_1)  
file_handler.close()
```

| | A | B | |
|---|-------------------------------------|---|--|
| 1 | bananacarrotavocadoorangegrapefruit | | |
| 2 | | | |
| 3 | | | |

```
file_handler = open('other.csv', 'w')  
list_1 = ('banana', 'carrot', 'avocado', 'orange', 'grapefruit')  
for item in list_1:  
    file_handler.write(item + '\n')  
file_handler.close()
```

| | A | |
|---|------------|--|
| 1 | banana | |
| 2 | carrot | |
| 3 | avocado | |
| 4 | orange | |
| 5 | grapefruit | |
| 6 | | |

File Open Modes

| Character | Meaning |
|-----------|---|
| 'r' | open for reading (default) |
| 'w' | open for writing, truncating the file first |
| 'x' | create a new file and open it for writing |
| 'a' | open for writing, appending to the end of the file if it exists |
| 'b' | binary mode |
| 't' | text mode (default) |
| '+' | open a disk file for updating (reading and writing) |
| 'U' | universal newline mode (deprecated) |

Removing a file

- Watch out! This removes the file **permanently!**

```
import os
os.remove("other.csv") #remove other.csv permanently!
```

The end