CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

http://www.cit.ie

Files and lists

Writing to a file

```
def write_to_file(numbers, filename):
    destination = open(filename, 'w')
    for i in range(len(numbers)):
        destination.write(str(numbers[i]) + "\n")
    destination.close()

def main():
    list_of_numbers = [5,2,1]
    output_filename = "output.txt"
    write_to_file(list_of_numbers, output_filename )

main()
```

write all numbers to the file using write() method.

List_of_numbers:

```
[5, 2, 1]
```

output.txt

```
5
2
1
```

Writing to a file - alternative

```
def write_to_file(numbers, filename):
    destination = open(filename, 'w')
    for i in range(len(numbers)):
        print(numbers[i], file=destination)
    destination.close()

def main():
    list_of_numbers = [5,2,1]
    output_filename = "output.txt"
    write_to_file(list_of_numbers, output_filename )

main()
```

use print to send data to the file

List_of_numbers:

```
[5, 2, 1]
```

output.txt

```
5
2
1
```

Problem: Reading the contents of a file to an array

- Reading from file that contain all strings or all numbers
- When we read **names.txt** we want to create a list of strings

names.txt

```
Michael
Minerva
Daffodil
Bugs
```

names

```
["Michael", "Minerva", "Daffodil", "Bugs" ]
```

- When we read **numbers.txt** we want to create a list of numbers

numbers.txt

```
"5"
"6"
"7"
"3"
"4"
"5"
```

numbers

```
[5, 6, 7, 3, 5, 5 ]
```

- We'll revisit using the method **readline()** to read one line at a time from the file. We create a list in each to the item to the list.

Warning: File contains string and at the end of each line is a new line character

Reading a list of numbers from a file – using readline()

```
def read_data(name_of_file):
    new_list = []
    connection = open(name_of_file)
    while True:
        line = connection.readline()
        if line == "":
            break
        x = int(line)
        new_list.append(x)
    return new_list


def main():
    data_file = "numbers.txt"
    list_of_numbers = read_data(data_file)
    print(list_of_numbers)

main()
```

Create an empty list

Open the file and store file handle/connection

Read one line at a time until an empty string is read

Convert each number to an int and add to our new_list

numbers.txt

```
5
6
7
3
4
5
```

new_list

```
[5, 6, 7, 3, 5, 5 ]
```

readlines() reads contents into an list (strings)


```
def read_data(name_of_file):
    connection = open(name_of_file)
    lines = connection.readlines()
    return lines

def main():
    filename = "names.txt"
    list_of_names = read_data(filename)
    print(list_of_names)

main()
```

readlines() creates a list of strings. Each line in the file becomes an item in the list.

File : names.txt

```
Michael
Minerva
Daffodil
Bugs
```

Output

```
["Michael", "Minerva", "Daffodil", "Bugs" ]
```

readlines() – read contents to list (numbers)

```
def read_data(name_of_file):
    connection = open(name_of_file)
    lines = connection.readlines()
    numbers = []
    for line in lines:
        numbers.append(int(line))
    return numbers

def main():
    filename = "numbers.txt"
    list_of_numbers = read_data(filename)
    print(list_of_numbers)

main()
```

Create a list of strings

Create a new list of numbers using a for loop

numbers.txt

```
5
6
7
3
4
5
```

lines

```
["5", "6", "7", "3", "5", "5" ]
```

numbers

```
[5, 6, 7, 3, 5, 5 ]
```

readlines() – read contents to list of numbers – using list comprehension


```
def read_data(name_of_file):
    connection = open(name_of_file)
    lines = connection.readlines()
    numbers = [int(line) for line in lines]
    return numbers

def main():
    filename = "numbers.txt"
    list_of_numbers = read_data(filename)
    print(list_of_numbers)

main()
```

Read a list of strings

Create a corresponding list of numbers using list comprehension

numbers.txt

```
5
6
7
3
4
5
```

lines

```
["5", "6", "7", "3", "5", "5" ]
```

numbers

```
[5, 6, 7, 3, 5, 5 ]
```

Problem. 2: Read from a line that contains comma separated fields

- What if all the data is on one line
- We need to read the line into a string using **readline()**
- Then create an list of individual strings by calling the string method **.split()**
- Then create a list of numbers using a for loop or by list comprehension technique.

Numbers_one_line.txt

```
"5,6,7,3,4,5"
```

line

```
"5,6,7,3,4,5"
```

data

```
["5", "6", "7", "3", "5", "5" ]
```

numbers

```
[5, 6, 7, 3, 5, 5 ]
```

Use readline() and split()

```
def read_data(name_of_file):
    connection = open(name_of_file)
    line = connection.readline()
    data = line.split(',')
    numbers = []
    for i in range(len(data)):
        numbers.append(int(data[i]))
    return numbers

def main():
    filename = "numbers_one_line.txt"
    list_of_numbers = read_data(filename)
    print(list_of_numbers)
```

Read a line contain , separated items

Use split method to create a list of these items

Create a list of numbers from this list

line

"5,6,7,3,4,5"

data

["5","6", "7", "3", "5", "5"]

Numbers_one_line.txt

"5,6,7,3,4,5"

numbers

[5, 6, 7, 3, 5, 5]

Use readline() and split() + list comprehension

```
def read_data(name_of_file):
    connection = open(name_of_file)
    line = connection.readline()
    data = line.split(',')
    numbers = [int(x) for x in data]
    return numbers

def main():
    filename = "numbers_one_line.txt"
    list_of_numbers = read_data(filename)
    print(list_of_numbers)
```

Replace loop with list comprehension

line

"5,6,7,3,4,5"

data

["5","6", "7", "3", "5", "5"]

Numbers_one_line.txt

"5,6,7,3,4,5"

numbers

[5, 6, 7, 3, 5, 5]

Problem 3:

- Sometimes the data will be of different types and spread over more than 1 line. In this example the name is on one line followed by the age of a person on the next line.
- We want to create two lists – the first line is a string and goes on the name list , the second line is a number and goes on the ages list and so on until the we have read the whole file.

```
Ann
5
Fred
6
Betty
7
Matilda
3
Michael
4
Minerva
5
```

names

```
["Ann","Fred", "Betty", "Matilda", "Michael", "Minerva" ]
```

ages

```
[5, 6, 7, 3, 5, 5 ]
```

Use readline() and split()

```
def read_data(name_of_file):
    connection = open(name_of_file)
    names_list = []
    ages_list = []
    while True:
       line = connection.readline().rstrip()
       if line == "":
            break
       names_list.append(line)
       line = connection.readline().rstrip()
       ages_list.append(int(line))
    return names_list, ages_list

def main():
    filename = "kids.txt"
    names, ages = read_data(filename)
    print(names)
    print(ages)
```

create two empty lists

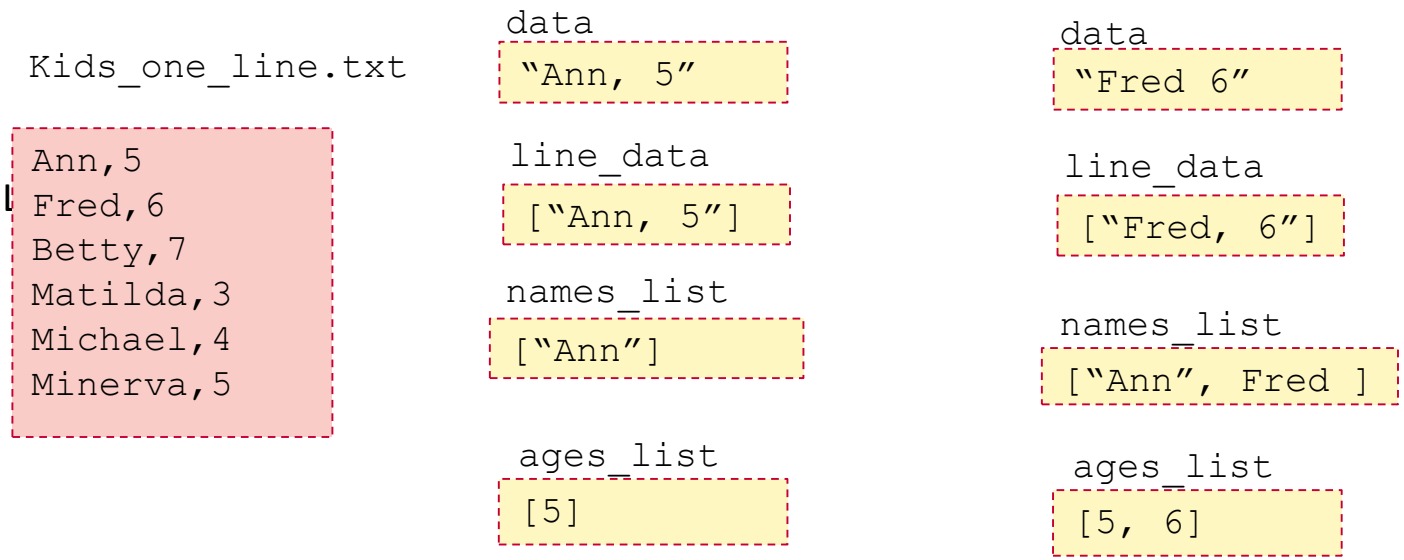
Read the first line in pair + add to the names

We know we are at then end of the file if the name is “” so break

Read the second line in the pair and add to the ages list – store as an int

Problem 4:

- The code to read the file depends on the layout of the file
- The file might contain all the data about one person on one line.
- Here we call **readline()** for each line and use split() to separate name and age. Then we add each to the correct list



Use readline() and split() + list comprehension

```
def read_data(name_of_file):
    connection = open(name_of_file)
    names_list = []
    ages_list = []
    while True:
        line = connection.readline()
        if line == "":
            break
        line_data = line.split(',')
        names_list.append(line_data[0])
        ages_list.append(int(line_data[1]))
    return names_list, ages_list

def main():
    filename = "kids_one_line.txt"
    names, ages = read_data(filename)
    print(names)
    print(ages)
```

line
"Ann, 5"

line_data
["Ann, 5"]

names_list
["Ann"]

ages_list
[5]