UNIVERSITY OF
GLOUCESTERSHIRE

CT4031 – MATHS FOR DATA SCIENCE

Strings

string: A sequence of text characters in a program.

Strings start and end with quotation mark `"` or apostrophe `'` characters.

Examples:

```
"hello"
"This is a string"
"This, too, is a string.   It can be very long!"
```

A string can represent characters by preceding them with a backslash.

`\t`	tab character
`\n`	new line character
`\"`	quotation mark character
`\\`	backslash character

Example: `"Hello\tthere\nHow are you?”`

Indexes

Characters in a string are numbered with *indexes* starting at 0:

Example:

```
name = "P. Parker”
```

index	0	1	2	3	4	5	6	7	8
character	P	.		P	a	r	k	e	r

Accessing an individual character of a string:

variableName [*index*]

```
name[0] => ‘P’
```

String properties

`len`(*string*)	- number of characters in a string (including spaces)
`str.lower`(*string*)	- lowercase version of a string
`str.upper`(*string*)	- uppercase version of a string

Example:

```
name = "Peter Parker"
length = len(name)
big_name = str.upper(name)
print (big_name, "has", length, "characters")
```

Output:

```
PETER PARKER has 12 characters
```

File processing

Many programs handle data, which often comes from files.

Reading the entire contents of a file:

variableName `= open("`*filename*`").read()`

Example:

```
file_text = open(”test.txt").read()
```

Line-by-line processing

```
for line in open("filename").readlines():
      statements
```

Example:

```
   count = 0
   for line in open(”test.txt").readlines():
       count = count + 1
   print ("The file contains", count, "lines.”)
```

Writing files

open ( ) will return a file object

Modes:

- ‘*r*’ – Read mode
- ‘*w*’ – Write mode
- ‘*a*’ – Appending mode

Writing files

```
f = open("testfile.txt","a")

f.write("Hello World\n")
f.write("This is our new text file")
f.write("and this is another line.")
f.write("Why? Because we can.")

f.close()
```

Task

- Write a program that reads the file students.txt
- Write a program that receives a new student and writes it on students.txt
- Write a program that calculates the average grade of each student and writes it in a new file

Functions

A *function* is a piece of code that performs a task of some kind.

A function has a name that is used when we need for the task to be executed. Asking that the task be executed is referred to as “calling” the function.

Some functions need one or more pieces of input when they are called.

Some functions give back a value. If a function gives back a value, this is referred to as “returning” the value.

Functions

Function definition begins with “def.”

Function name and its arguments.

```
def get_final_answer(filename):
    line1
    line2
    return total_counter
```

Colon.

The indentation matters…
First line with less
indentation is considered to be
outside of the function definition.

The keyword ‘return’ indicates the
value to be sent back to the caller.

Exercises

1) Write a Python function to reverse a string

2) Write a Python function to multiply all the numbers in a list.

Sample List : (8, 2, 3, -1, 7)

Expected Output : -336