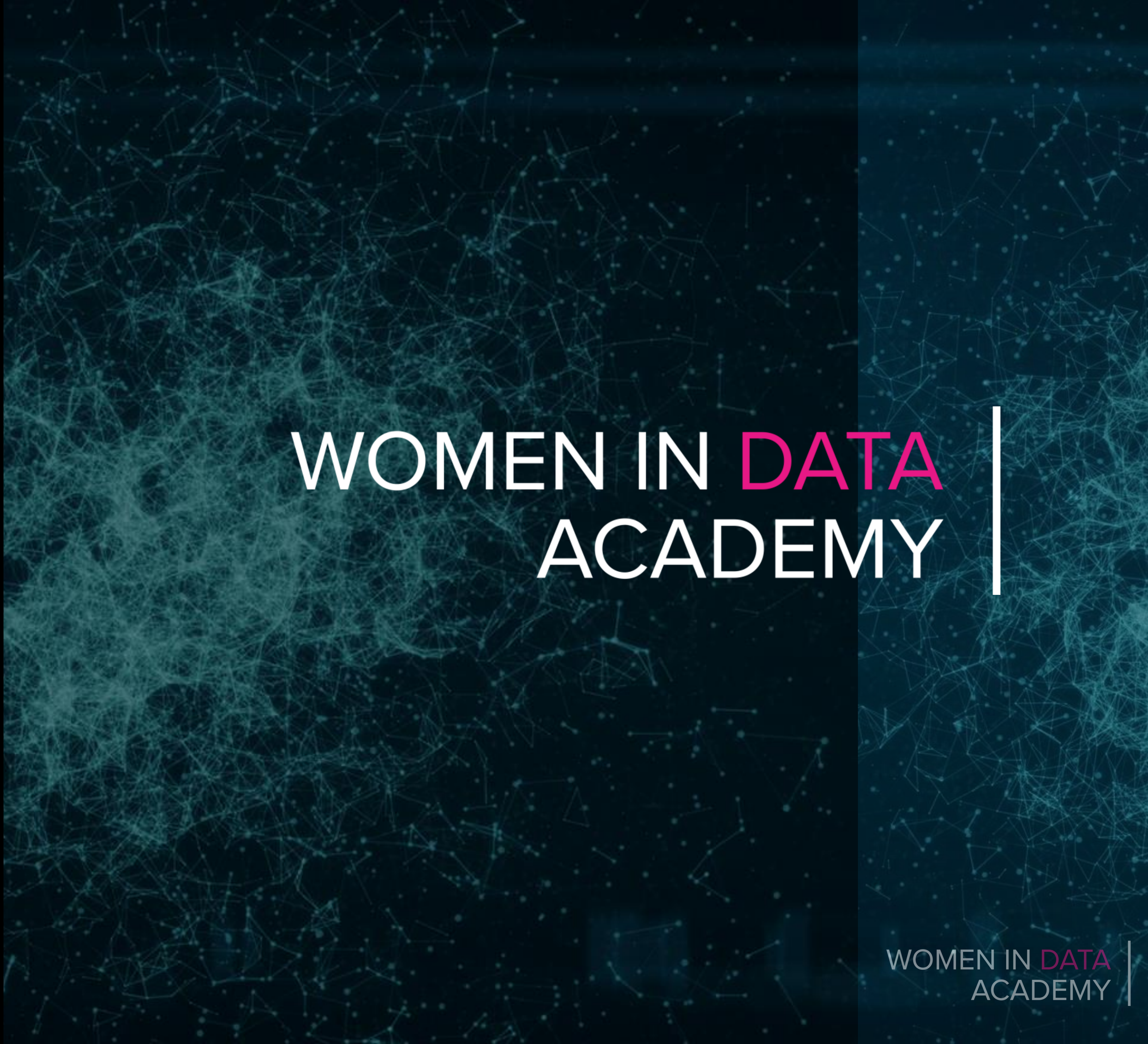
Data Science Python

Module 1 – Session 3

TECH TALENT ACADEMY

WOMEN IN DATA ACADEMY



Session Content

1+2 DIMENSIONAL LISTS

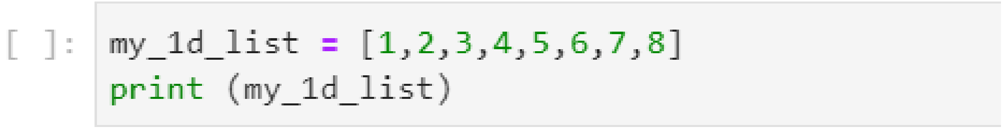
STRING HANDLING

FILE HANDLING

WOMEN IN DATA ACADEMY

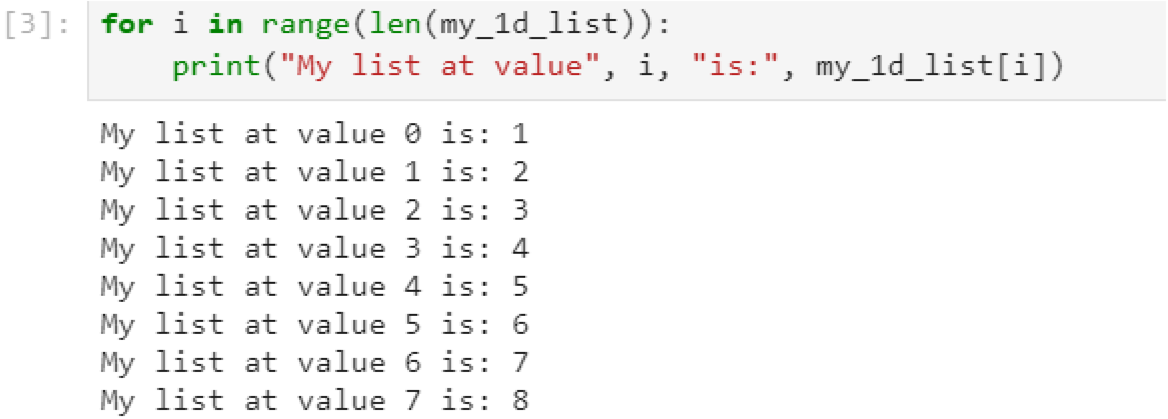
Lists

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8



```
[ ]: my_1d_list = [1,2,3,4,5,6,7,8]
     print (my_1d_list)
```

Python is zero indexed



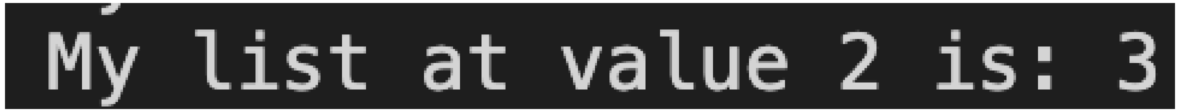
```
[3]: for i in range(len(my_1d_list)):
         print("My list at value", i, "is:", my_1d_list[i])

     My list at value 0 is: 1
     My list at value 1 is: 2
     My list at value 2 is: 3
     My list at value 3 is: 4
     My list at value 4 is: 5
     My list at value 5 is: 6
     My list at value 6 is: 7
     My list at value 7 is: 8
```

Handling 1D Lists

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8

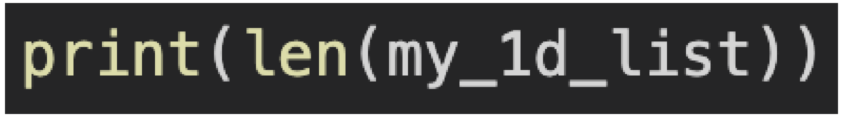
1dlist = [1,2,3,4,5,6,7,8]

print (1dlist[2])



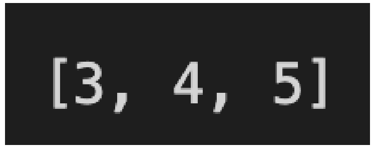
```
My list at value 2 is: 3
```

print (len(1dlist))



```
print(len(my_1d_list))
```

print (1dlist[2:5])

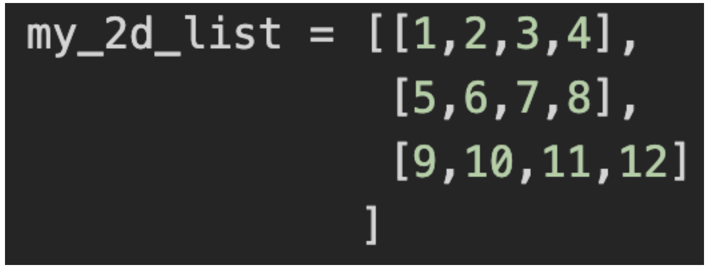


```
[3, 4, 5]
```

Handling 2D Lists

2-Dimensional Lists are lists within a list

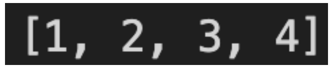
This is done by putting [] within [] and separating by ,



```
my_2d_list = [[1,2,3,4],
              [5,6,7,8],
              [9,10,11,12]
             ]
```

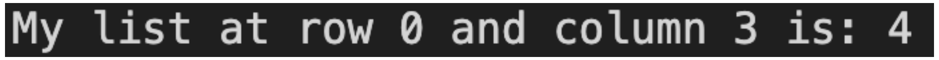
They are referenced as row then column.

2dlist[0] will show you the content of the row



```
[1, 2, 3, 4]
```

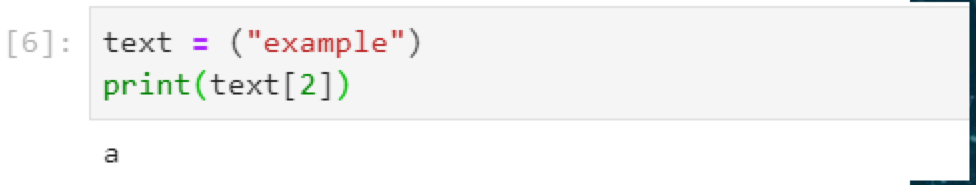
2dlist[0][3] will show you the first row, and 4th column/value



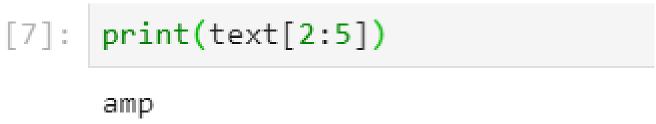
```
My list at row 0 and column 3 is: 4
```

String Handling

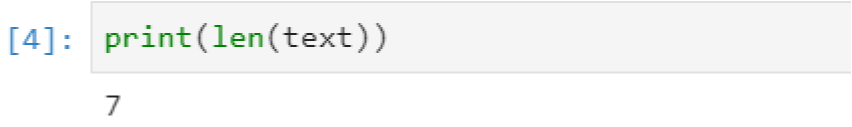
E	x	a	m	p	l	e	s
0	1	2	3	4	5	6	7



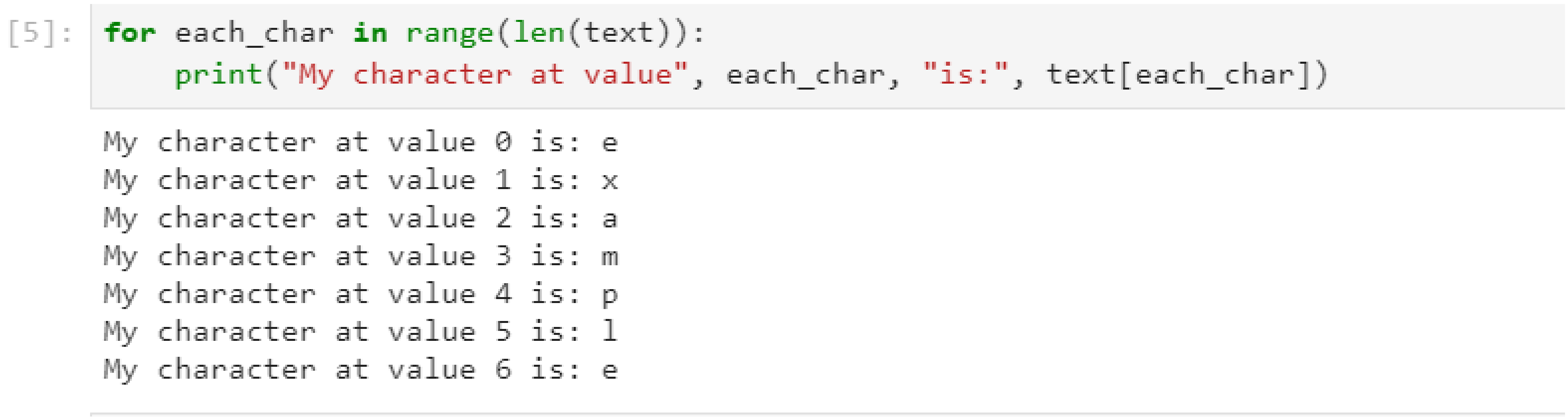
```
[6]: text = ("example")
     print(text[2])

     a
```



```
[7]: print(text[2:5])

     amp
```



```
[4]: print(len(text))

     7
```



```
[5]: for each_char in range(len(text)):
         print("My character at value", each_char, "is:", text[each_char])

     My character at value 0 is: e
     My character at value 1 is: x
     My character at value 2 is: a
     My character at value 3 is: m
     My character at value 4 is: p
     My character at value 5 is: l
     My character at value 6 is: e
```

WOMEN IN DATA
ACADEMY

Storing Data

When a program is run the data is stored in memory, unless it is held in permanent storage it will be lost.

FLAT FILES

Simple text files   myfile.txt

Use notepad to open them up (note that both the text and python files need to be in the same folder)

File Handling

To create a .txt file we use open()

```
my_file = open("my_new_text_file.txt", “w")
```

This will create a text file called “my_new_text_file”.

If the file already exists it will overwrite the data!

File modes:

w = Writes new data to a file

r = Reads from a file

a = Append data to the end of the file

Writing to a file

File handle

Open a file to Write

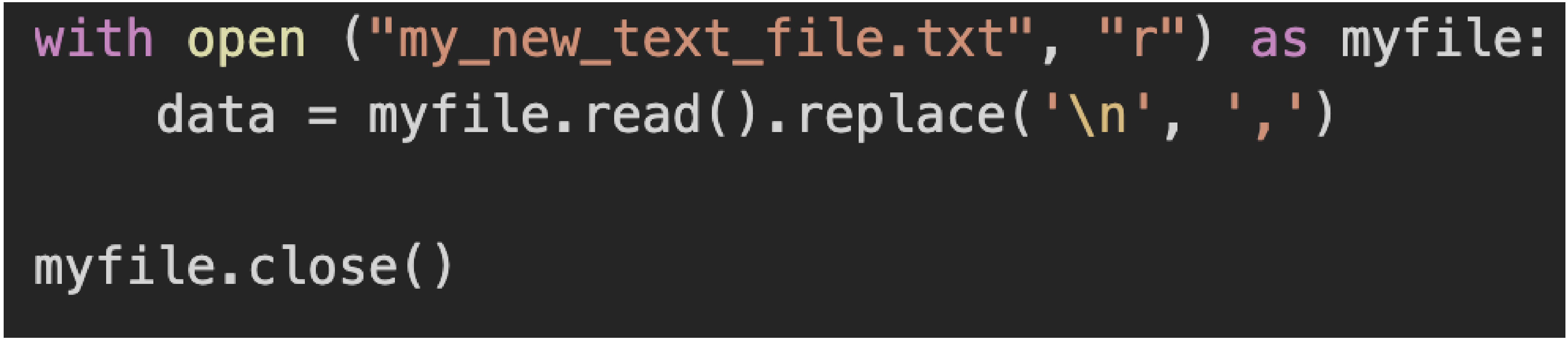
```
my_file = open("my_new_text_file.txt", “w")
for i in range(10):
    my_file.write(str(i))
    my_file.write("\n")

my_file.close()
```

New line operator

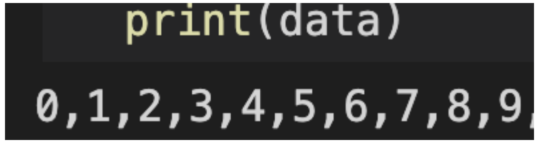
Files must be closed

WOMEN IN DATA
ACADEMY

Reading from a file

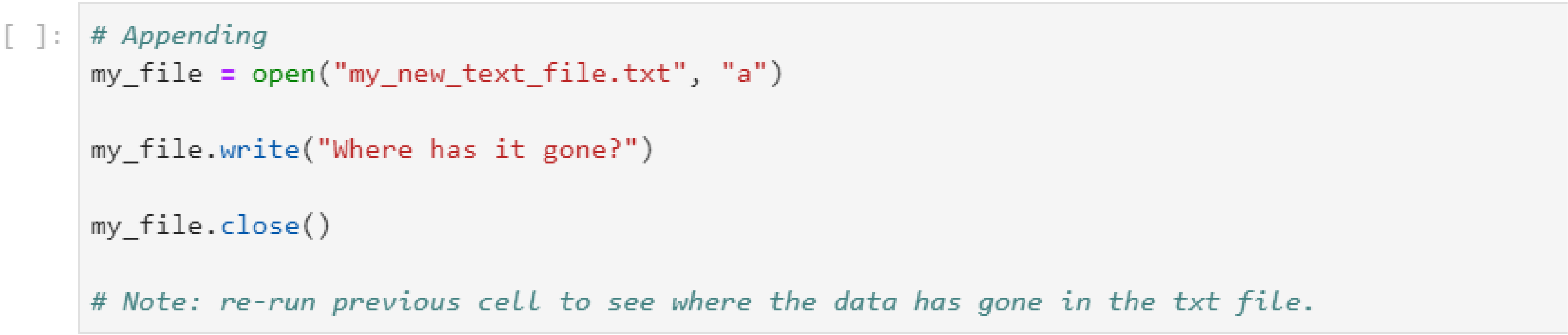


```
with open ("my_new_text_file.txt", "r") as myfile:
    data = myfile.read().replace('\n', ',')

myfile.close()
```



```
    print(data)
0,1,2,3,4,5,6,7,8,9,
```

Appending to a file



```
[ ]: # Appending
     my_file = open("my_new_text_file.txt", "a")

     my_file.write("Where has it gone?")

     my_file.close()

     # Note: re-run previous cell to see where the data has gone in the txt file.
```

Home Learning Tasks

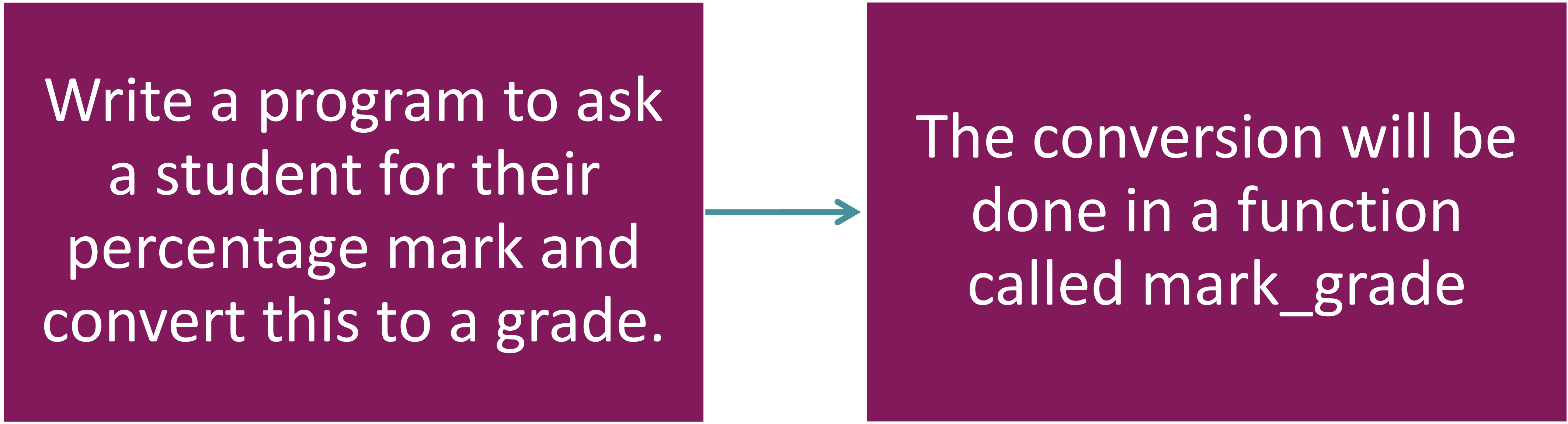
Task 1

Write a program that allows you to enter 4 numbers and stores them in a file called “Numbers”

- **3**
- **45**
- **83**
- **21**

Have a go at ‘w’ ‘r’ ‘a’

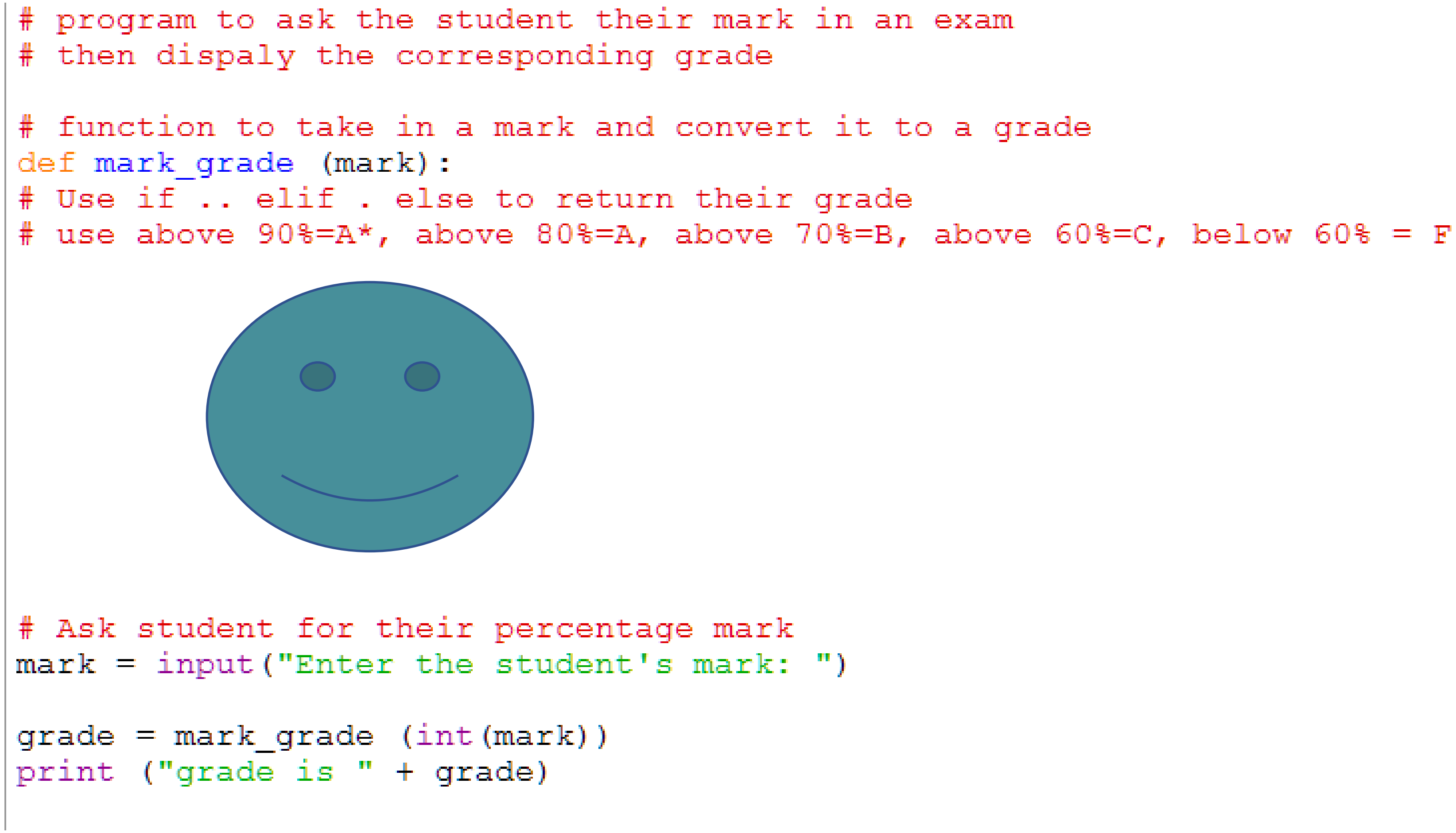
Task 2



Write a program to ask a student for their percentage mark and convert this to a grade.

The conversion will be done in a function called mark_grade

Answer



```
# program to ask the student their mark in an exam
# then dispaly the corresponding grade

# function to take in a mark and convert it to a grade
def mark_grade (mark):
# Use if .. elif . else to return their grade
# use above 90%=A*, above 80%=A, above 70%=B, above 60%=C, below 60% = F





# Ask student for their percentage mark
mark = input("Enter the student's mark: ")

grade = mark_grade (int(mark))
print ("grade is " + grade)
```

Extension to Task 2

- Ask the user for their target grade and print this with their mark
- If their target grade > exam grade display a suitable message
- If their target grade = exam grade display a suitable message
- If their target grade < exam grade display a suitable message

WOMEN IN DATA
ACADEMY

WOMEN IN DATA
ACADEMY