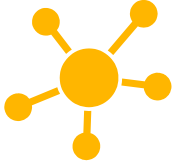




Recommending with **Data**

Working with files.



Review Questions


What is the most basic algorithm for a recommendation system?

What is a more advanced algorithm for a recommendation system?

What will this code output?

```
print(5//3)
print(2**3)
print(10%3)
```

Big(ish) Data

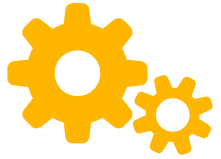


Hooray! Now we don't have to put all our strings and data directly in our code.

We're going to use big pieces of data, from a file.

The goal is to make a program that can analyse this data and make a few comments about it.





This lesson

- Splitting strings into lists (**split**)
- Opening files (**open**)
- Reading files
- Accessing specific elements of a **list** (**indexing** with [])
- Comparing numbers



Review of Readings

What will this code output?

```
weather = ["sunny", "rainy", "cloudy"]  
print(weather[0])  
print(weather[2])
```

What will this code output?

```
today = "sunny"  
print(today[0])  
print(today[2])
```

A Survey

Who are you? (Please provide a distinctive, memorable *fake* name) *

Your answer _____

What is your favourite movie genre? *

- Comedy
- Horror
- Drama
- Action
- Fantasy
- Sci-fi
- Animated

Favourite animal as a pet? *

- Cat
- Dog
- Bird
- Fish
- Frog
- Rodent
- Insect
- Snake
- Turtle
- Spider
- Other: _____

Favourite world cuisine? *

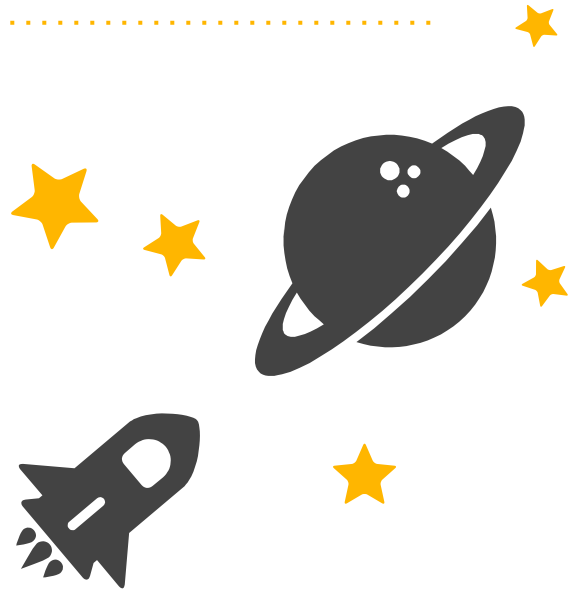
- Italian
- Japanese
- Korean
- Chinese
- Malaysian
- Indian
- American/Canadian
- Spanish
- Thai
- Turkish
- Greek
- Other: _____

Favourite hobby? *

- Playing video games
- Playing an instrument
- Dancing
- Painting or drawing
- Playing a sport
- Working out
- Crafts
- Acting
- Singing
- Yoga
- Making videos
- Cooking
- Learning new languages

Files

Where can we keep lots of data?





Add a **data file**

Step 1

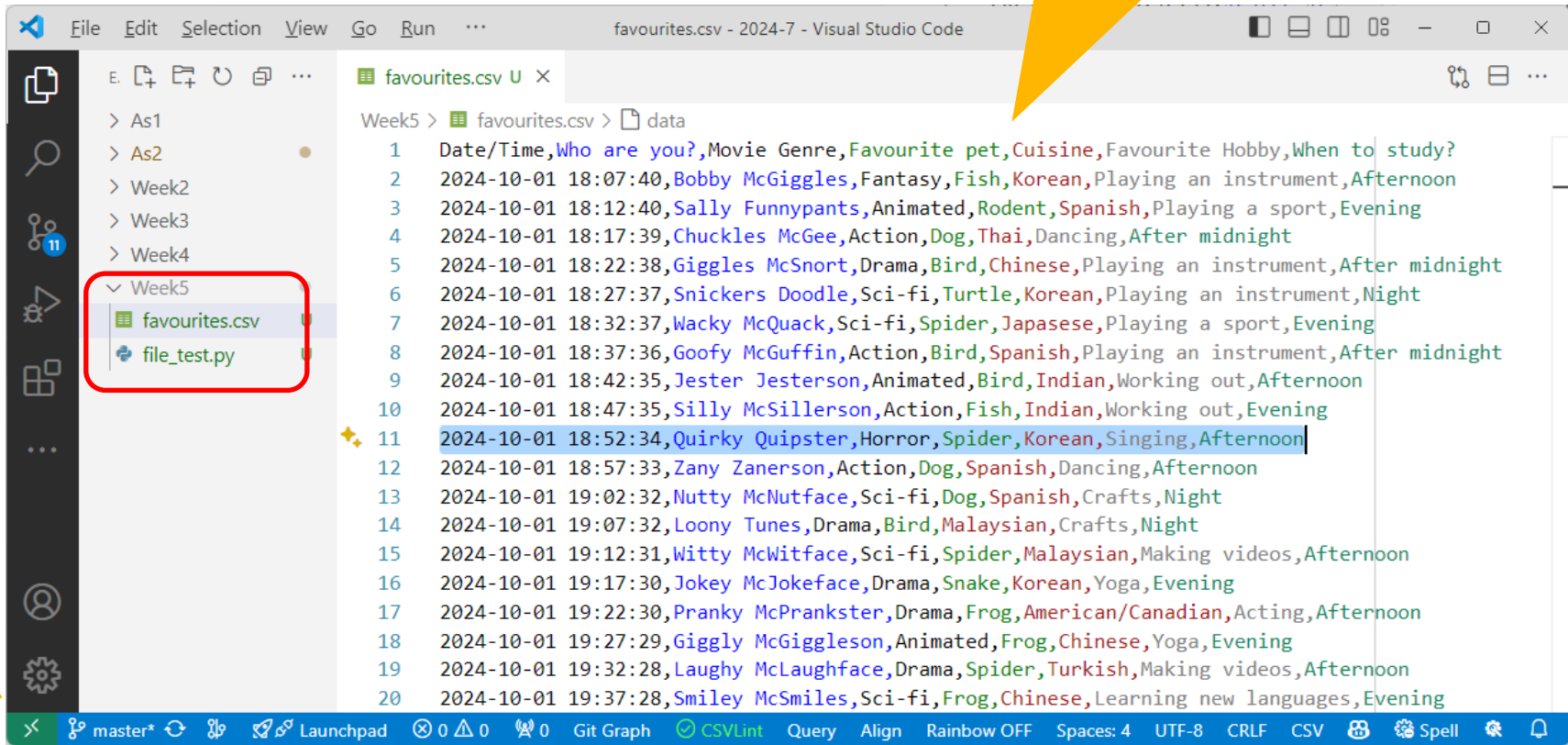
Download **favourites.csv** from course website

Step 2

Place in **same folder/location** as your Python (.py) file on your PC

Add a data file

I have the "Rainbow CSV"
VS Code extension.



```
Week5 > favourites.csv > data
1 Date/Time,Who are you?,Movie Genre,Favourite pet,Cuisine,Favourite Hobby,When to study?
2 2024-10-01 18:07:40,Bobby McGiggles,Fantasy,Fish,Korean,Playing an instrument,Afternoon
3 2024-10-01 18:12:40,Sally Funnypants,Animated,Rodent,Spanish,Playing a sport,Evening
4 2024-10-01 18:17:39,Chuckles McGee,Action,Dog,Thai,Dancing,After midnight
5 2024-10-01 18:22:38,Giggles McSnort,Drama,Bird,Chinese,Playing an instrument,After midnight
6 2024-10-01 18:27:37,Snickers Doodle,Sci-fi,Turtle,Korean,Playing an instrument,Night
7 2024-10-01 18:32:37,Wacky McQuack,Sci-fi,Spider,Japasese,Playing a sport,Evening
8 2024-10-01 18:37:36,Goofy McGuffin,Action,Bird,Spanish,Playing an instrument,After midnight
9 2024-10-01 18:42:35,Jester Jesterson,Animated,Bird,Indian,Working out,Afternoon
10 2024-10-01 18:47:35,Silly McSillerson,Action,Fish,Indian,Working out,Evening
11 2024-10-01 18:52:34,Quirky Quipster,Horror,Spider,Korean,Singing,Afternoon
12 2024-10-01 18:57:33,Zany Zanerson,Action,Dog,Spanish,Dancing,Afternoon
13 2024-10-01 19:02:32,Nutty McNutface,Sci-fi,Dog,Spanish,Crafts,Night
14 2024-10-01 19:07:32,Loony Tunes,Drama,Bird,Malaysian,Crafts,Night
15 2024-10-01 19:12:31,Witty McWitface,Sci-fi,Spider,Malaysian,Making videos,Afternoon
16 2024-10-01 19:17:30,Jokey McJokeface,Drama,Snake,Korean,Yoga,Evening
17 2024-10-01 19:22:30,Pranky McPrankster,Drama,Frog,American/Canadian,Acting,Afternoon
18 2024-10-01 19:27:29,Giggly McGiggleson,Animated,Frog,Chinese,Yoga,Evening
19 2024-10-01 19:32:28,Laughy McLaughface,Drama,Spider,Turkish,Making videos,Afternoon
20 2024-10-01 19:37:28,Smiley McSmiles,Sci-fi,Frog,Chinese,Learning new languages,Evening
```



Dog vs. Cat

```
1 Date/Time,Who are you?,Movie Genre,Favourite pet,Cuisine,Favourite Hobby,When to study?
2 2024-10-01,Bobby McGiggles,Animated,Dog,Korean,Crafts,Evening
3 2024-10-01,Sally Funnypants,Action,Dog,Chinese,Crafts,Afternoon
4 2024-10-01,Chuckles McGee,Sci-fi,Insect,Italian,Yoga,After midnight
5 2024-10-01,Giggles McSnort,Sci-fi,Frog,Korean,Panting or drawing,Night
6 2024-10-01,Snickers Doodle,Action,Dog,Spanish,Working out,Night
7 2024-10-01,Wacky McQuack,Action,Fish,Japasese,Yoga,Evening
8 2024-10-01,Goofy McGuffin,Fantasy,Turtle,Chinese,Panting or drawing,Evening
9 2024-10-01,Jester Jesterson,Comedy,Cat,Japasese,Crafts,Night
10 2024-10-01,Silly McSillerson,Fantasy,Dog,Turkish,Dancing,After midnight
11 2024-10-01,Quirky Quipster,Sci-fi,Fish,Thai,Acting,Afternoon
12 2024-10-01,Zany Zanerson,Animated,Turtle,Japasese,Crafts,Afternoon
13 2024-10-01,Nutty McNutface,Horror,Cat,Spanish,Acting,Morning
14 2024-10-01,Loony Tunes,Horror,Frog,Indian,Working out,Night
15 2024-10-01,Witty McWitface,Action,Cat,Turkish,Working out,Night
16 2024-10-01,Jokey McJokeface,Comedy,Cat,Korean,Singing,After midnight
17 2024-10-01,Pranky McPrankster,Animated,Spider,American/Canadian,Playing a sport,Evening
18 2024-10-01,Giggly McGiggleson,Animated,Snake,Chinese,Crafts,After midnight
19 2024-10-01,Laughy McLaughface,Comedy,Cat,Japasese,Cooking,After midnight
20 2024-10-01,Smiley McSmiles,Action,Dog,Korean,Singing,Night
```

Let's start by trying to find the **most popular animal as a pet.**



Reading Data from a File

- Must first **open** the CSV file.
- Then **read data, line-by-line**.
- Process each **line of data**.

Pointer to the current line

```
1 Date/Time,Who are you?,Movie Genre,Favourite pet,Cuisine,Favourite Hobby,When
2 2024-10-01,Bobby McGiggles,Animated,Dog,Korean,Crafts,Evening
3 2024-10-01,Sally Funnypants,Action,Dog,Chinese,Crafts,Afternoon
4 2024-10-01,Chuckles McGee,Sci-fi,Insect,Italian,Yoga,After midnight
5 2024-10-01,Giggles McSnort,Sci-fi,Frog,Korean,Panting or drawing,Night
6 2024-10-01,Snickers Doodle,Action,Dog,Spanish,Working out,Night
7 2024-10-01,Wacky McQuack,Action,Fish,Japasese,Yoga,Evening
8 2024-10-01,Goofy McGuffin,Fantasy,Turtle,Chinese,Panting or drawing,Evening
9 2024-10-01,Jester Jesterson,Comedy,Cat,Japasese,Crafts,Night
10 2024-10-01,Silly McSillerson,Fantasy,Dog,Turkish,Dancing,After midnight
11 2024-10-01,Quirky Quipster,Sci-fi,Fish,Thai,Acting,Afternoon
12 2024-10-01,Zany Zanerson,Animated,Turtle,Japasese,Crafts,Afternoon
13 2024-10-01,Nutty McNutface,Horror,Cat,Spanish,Acting,Morning
14 2024-10-01,Loony Tunes,Horror,Frog,Indian,Working out,Night
15 2024-10-01,Witty McWitface,Action,Cat,Turkish,Working out,Night
16 2024-10-01,Jokey McJokeface,Comedy,Cat,Korean,Singing,After midnight
17 2024-10-01,Pranky McPrankster,Animated,Spider,American/Canadian,Playing a sport
18 2024-10-01,Giggly McGiggleson,Animated,Snake,Chinese,Crafts,After midnight
19 2024-10-01,Laughy McLaughface,Comedy,Cat,Japasese,Cooking,After midnight
20 2024-10-01,Smiley McSmiles,Action,Dog,Korean,Singing,Night
```

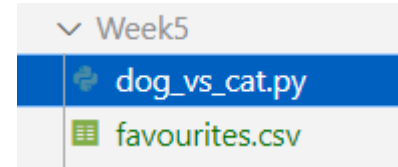
Get the data

What is the next line?

```
1 Date/Time,Who are you?,Movie Genre,Favourite pet,Cuisine,Favourite Hobby,
2 2024-10-01,Bobby McGiggles,Animated,Dog,Korean,Crafts,Evening
3 2024-10-01,Sally Funnypants>Action,Cat,Chinese,Crafts,Afternoon
```

The first line is just a header (not data)

```
1 # Find out if cats are more popular than dogs!
2 import pathlib
3
4 # Open the data file
5 # We have to tell Python where to find the file, and the file's name.
6 # a) Get name of folder where this code is saved
7 folder_of_code = pathlib.Path(__file__).parent.resolve()
8 # b) Build the full name of the `favourites.csv` file in that folder
9 full_file_name = f"{folder_of_code}/favourites.csv"
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Read (and print) the first line of data
17 first_line = my_csv_file.readline()
18 print(f"First line of data: {first_line}")
```



open() needs to find the file.
Build the full path based on where
our Python .py code is.

The **readline()** function reads and
moves the current line pointer to the
next line in the file object

data is now a **string**
corresponding to the 2nd line

Get the Data



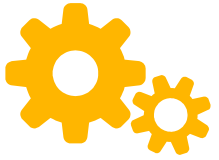
```
1 # Find out if cats are more popular than dogs!
2 import pathlib
3
4 # Open the data file
5 # We have to tell Python where to find the file
6 # a) Get name of folder where this code is saved
7 folder_of_code = pathlib.Path(__file__).parent.resolve()
8 # b) Build the full name of the `favourites.csv` file in that folder
9 full_file_name = f"{folder_of_code}/favourites.csv"
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Read (and print) the first line of data
17 first_line = my_csv_file.readline()
18 print(f"First line of data: {first_line}")
```

After 2
readline()'s

```
1 Date/Time,Who are you?,Movie Genre,Favourite pet,Cuisine,Favourite H
2 2024-10-01,Bobby McGiggles,Animated,Dog,Korean,Crafts,Evening
3 2024-10-01,Sally Funnypants,Action,Dog,Chinese,Crafts,Afternoon
4 2024-10-01,Chuckles McGee,Sci-fi,Insect,Italian,Yoga,After midnight
5 2024-10-01,Giggles McSnort,Sci-fi,Frog,Korean,Painting or drawing,Nig
6 2024-10-01,Snickers Doodle,Action,Dog,Spanish,Working out,Night
```

Looks good, let's keep going!
How do we get just the favourite pet part?

First line of data: 2024-10-01,Bobby McGiggles,Animated,Dog,Korean,Crafts,Evening



String operations (cont'd)

Example

Split up a string at each comma.
It returns a list of the items (without
commas)

```
myString.split(",")
```

Accessing the first element of the list

```
foods[0]
```



Splitting strings into lists

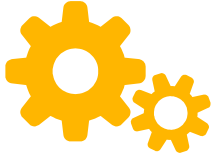
```
1 # Find out if cats are more popular than dogs!
2 import pathlib
3
4 # Open the data file
5 # We have to tell Python where to find the file, and the file's name.
6 # a) Get name of folder where this code is saved
7 folder_of_code = pathlib.Path(__file__).parent.resolve()
8 # b) Build the full name of the `favourites.csv` file in that folder
9 full_file_name = f"{folder_of_code}/favourites.csv"
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Read the first line of data
17 first_line = my_csv_file.readline()
18
19 # Split the first line of data into columns (using the comma)
20 columns = first_line.split(",")
21 print(f"Line of Data: {first_line}")
22 print(f"Columns:      {columns}")
```

Use **split** to

- 1) **split** up a **string** into its **elements** and
- 2) **return** the **elements** as a **list**



Now we have a list stored in **datalist**



The Index Operator

We can get a specific element in the list! The first, or second, or third...

Example

Creating a list

```
foods = ["cherries", "tomatoes"]
```

Accessing the first element of the **list**

```
foods[0]
```

These square brackets are called the **index** operator.
foods[0] → "cherries"

Accessing elements of a **string**

```
veggie = "pickle"
```

```
veggie[3] → "k"
```

```
foods[0][0] →
```

```
foods[0][1] →
```




Accessing list items

```
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Read the first line of data
17 first_line = my_csv_file.readline()
18
19 # Split the first line of data into columns (a list)
20 columns = first_line.split(",")
21 print(f"Line of Data: {first_line}")
22 print(f"Columns:      {columns}")
23
24 # Access the item (column) we want:
25 # 4th column (index 3) is the favourite pet
26 print(f"Fav Pet: {columns[3]}")
```

We can access a specific element of the list using []. Remember: It starts from 0!

0 1 2 3 4 5 6

```
Columns:      ['2024-10-01', 'Bobby McGiggles', 'Animated', 'Dog', 'Korean', 'Crafts', 'Evening\n']
Fav Pet: Dog
```



Looping over lines in a file

```
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Print all pet choices
17 for line in my_csv_file:
18     # Convert line of data into a list of columns
19     columns = line.split(",")
20
21     # Access the item (column) we want:
22     # 4th column (index 3) is the favourite pet
23     print(f"Fav Pet: {columns[3]}")
24
```

We can go through the file line by line, using a **for-line-in-file-loop**. This is better than using `readline` in a loop because we don't know beforehand how many lines are in the file.



Get pet survey results

```
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Count number of dog vs cat people
17 dog_tally = 0
18 cat_tally = 0
19
20 # Read through the entire (remaining) file one line at a time
21 for next_line in my_csv_file:
22
23     # Get favourite pet from line
24     columns = next_line.strip().split(",")
25     fav_pet = columns[3]
26
27     # Check favourite pet
28     if fav_pet == "Dog":
29         dog_tally += 1
30     elif fav_pet == "Cat":
31         cat_tally += 1
32
33 # Print interesting
34 print(f"# dog people: {dog_tally}")
35 print(f"# cat people: {cat_tally}")
```

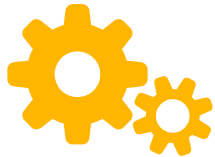
We know how to do this
from last time

We know how to do this
from last time

Comparing



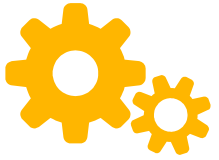
Comparing numbers! More than just ==



A common mistake

Common Mistake!

There is a very common mistake that occurs when programmers try to write boolean expressions. For example, what if we have a variable `number` and we want to check to see if its value is 5,6, or 7. In words we might say: “number equal to 5 or 6 or 7”. However, if we translate this into Python, `number == 5 or 6 or 7`, it will not be correct. The `or` operator must join the results of three equality checks. The correct way to write this is `number == 5 or number == 6 or number == 7`. This may seem like a lot of typing but it is absolutely necessary. You cannot take a shortcut.



Review: Comparison operators

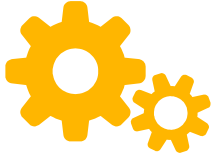
	Example
myScore does not equal 5	<code>myScore != 5</code>
myScore is greater than 5	<code>myScore > 5</code>
myScore is less than 5	<code>myScore < 5</code>
myScore is greater than or equal to 5	<code>myScore >= 5</code>

Result analyses

```
1 # Find out if cats are more popular than dogs!
2 import pathlib
3
4 # Open the data file
5 # We have to tell Python where to find the file, and the
6 # a) Get name of folder where this code is saved
7 folder_of_code = pathlib.Path(__file__).parent.resolve()
8 # b) Build the full name of the `favourites.csv` file in the
9 full_file_name = f"{folder_of_code}/favourites.csv"
10 # c) Open the file
11 my_csv_file = open(full_file_name)
12
13 # Skip over the header (column names)
14 unused_header_line = my_csv_file.readline()
15
16 # Count number of dog vs cat people
17 dog_tally = 0
18 cat_tally = 0
19
```

We can use **comparison operators** along with **logical operators** (and/or)

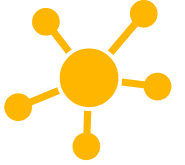
```
20 # Read through the entire (remaining) file one line at a time
21 for next_line in my_csv_file:
22
23     # Get favourite pet from line
24     columns = next_line.strip().split(",")
25     fav_pet = columns[3]
26
27     # Check favourite pet
28     if fav_pet == "Dog":
29         dog_tally += 1
30     elif fav_pet == "Cat":
31         cat_tally += 1
32
33 # Print interesting
34 print(f"# dog people: {dog_tally}")
35 print(f"# cat people: {cat_tally}")
36
37
38 # Print something interesting
39 if cat_tally < 1:
40     print("Nobody loves cats! :( ")
41 elif cat_tally > dog_tally and cat_tally < 30:
42     print("More cat people than dog people, but not many total")
```



Operator Precedence

Level	Category	Operators
7(high)	exponent	**
6	multiplication	*,/,//,%
5	addition	+,-
4	relational	==,!=,<=,>=,>,<
3	logical	not
2	logical	and
1(low)	logical	or

Let's **review** some concepts



How would you open a file called "survey.txt"?

How do you access the second to last element in a list called `favourites`?

What are 2 ways to read line(s) from a file?

How do you split a string of words into a list? Imagine that the words are separated by `;`

If a list contains the following:

```
singers = ["elsa",  
"anna", "snowman"]
```

Then who is in `singers[2]`?