

Recap: What is output at the end?

```
colours=[]
colours.append("red")
colours.append("green")
colours.append("blue")
colours.append("cyan")
colours.append("magenta")
colours.remove("cyan")
colours.pop(2)
print(colours[1])
print(colours[2])
```

Two dimensional lists

Learning objective: Explain and use two-dimensional lists

Two dimensional lists

Two dimensional lists are a more complex data structure that essentially involve lists of lists. Let us consider the a data structure that a library might have that includes the author name, title and publisher of a list of books.

Author	Book	Publisher
JK Rowling	Harry Potter and the Chamber of Secrets	Bloomsbury
Roald Dahl	The BFG	Penguin
Michael Morpurgo	Private Peaceful	HarperCollins
Julia Donaldson	The Gruffalo	Macmillan
Eric Carle	The Very Hungry Caterpillar	Penguin

Books 2D list

To start with let us create 3 one-dimensional lists.

```
authors=["JK Rowling", "Roald Dahl", "Michael Morpurgo","Julia Donaldson", "Eric Carle"]  
publisher=["Bloomsbury", "Penguin", "HarperCollins", "Macmillan", "Penguin"]  
title=["Harry Potter and the Chamber of Secrets", "The BFG", "Private Peaceful", "The  
Gruffalo", "The Very Hungry Caterpillar"]
```

We create an empty list in which we will add the authors, publisher, and title lists'

```
book=[]  
book.append(authors)  
book.append(publisher)  
book.append(title)  
  
print(book)
```

So this is how the books 2D table looks now.

```
[[ 'JK Rowling', 'Roald Dahl', 'Michael Morpurgo', 'Julia Donaldson', 'Eric Carle'],  
 [ 'Bloomsbury', 'Penguin', 'HarperCollins', 'Macmillan', 'Penguin'],  
 [ 'Harry Potter and the Chamber of Secrets', 'The BFG', 'Private Peaceful', 'The Gruffalo', 'The Very Hungry Caterpillar']]
```

To find the index position of an item we can use.

```
book[1].index("Bloomsbury")
```

This returns 0.

Books 2D list

Let us see not how to access the different elements in the 2D list. We can think of the 2D list as a grid or table, with index positions in two directions horizontally and vertically. The index position starts at zero, although normally in the exam questions it starts a 1. (Make sure that you read the question).

- The position in the list of the author JK Rowling is 0,0: `books[0][0]`
- The position in the list of the Publisher HarperCollins is 1,2. `books[1][2]`
- It is easy to get confused as to which way around the index goes. The pattern is:

```
list_2d[list][position_of_item_in_list]
```

Index position	0	1	2	3	4
0	JK Rowling	Roald Dahl	Michael Morpurgo	Julia Donaldson	Eric Carle
1	Bloomsbury	Penguin	HarperCollins	Macmillan	Penguin
2	Harry ..	The BFG	Private Peaceful	The Gruffalo	The Very ..

Task books?

1. What index positions is the Publisher Penguin at?
2. What is at index position books[2][2]?
3. What is at index position books[1][1]?
4. What is at index position books[0][4]?

Index position	0	1	2	3	4
0	JK Rowling	Roald Dahl	Michael Morpurgo	Julia Donaldson	Eric Carle
1	Bloomsbury	Penguin	HarperCollins	Macmillan	Penguin
2	Harry ..	The BFG	Private Peaceful	The Gruffalo	The Very ..

Same task but using the Python syntax instead of the table?

1. What index positions is the Publisher Bloomsbury at?
2. What is at index position books[1][2]?
3. What is at index position books[2][1]?
4. What is at index position books[0][2]?

```
[['JK Rowling', 'Roald Dahl', 'Michael Morpurgo', 'Julia Donaldson', 'Eric Carle'],  
 ['Bloomsbury', 'Penguin', 'HarperCollins', 'Macmillan', 'Penguin'],  
 ['Harry Potter and the Chamber of Secrets', 'The BFG', 'Private Peaceful', 'The  
Gruffalo', 'The Very Hungry Caterpillar']]
```


Task: What are the values returned in each case for the two dimensional list a?

```
a = [ [23, 14, 17, 19, 21],  
      [12, 18, 37, 42, 1],  
      [16, 67, 83, 92, 4],  
      [55, 43, 76, 5, 10],  
      [78, 98, 3, 6, 75] ]
```

1. a[0][0]
2. a[2][2]
3. a[4][4]
4. a[3][1]
5. a[1][3]
6. a[4][2]
7. a[0][2]
8. What position is 4 in?
9. What position is 6 in?
10. a[4][5]

Presenting 2D lists

Unhelpfully a two dimensional could be presented as

```
a = [ [23, 14, 17], [12, 18, 37], [16, 67, 83]]
```

Which makes finding the index position of individual elements tricky, but it is easy enough to convert into a grid which makes interpretation easier.

```
a = [ [23, 14, 17], \  
      [12, 18, 37], \  
      [16, 67, 83]]
```

Task: phone and email app

- Write a program that contains a 2D list of 5 of your friends names along with their phone numbers and emails (which you can just make up)
- The user then enters the name of their friend and their phone number and email is returned.
- Extension: Adapt the program so that you can add more names

Name	Fred	Egbert	Dibble	Rudolph	Bart
Phone	01234567	89101112	13141516	17181920	21222324
Email	fred@gmail.com	egbert@gmail.com	dibble@gmail.com	rudolph@gmail.com	bart@gmail.com

Task: Time table app

- Write a program that contains a 2d list of your time table

	1	2	3	4	5	6
M	French	French	Maths	Maths	English	English
Tu						
W						
Th						
F						

- The user then enters the day of the week and the period and the lesson at that time is returned.

Task: What are the values returned in each case for the two dimensional list a?

```
y = [ [q,w,e,r,t],  
      [u,i,o,p,a],  
      [s,d,f,g,h],  
      [j,k,l,z,x],  
      [c,v,b,n,m] ]
```

1. y[0][0]
2. y[1][2]
3. y[2][1]
4. y[3][3]
5. y[1][3]
6. y[2][4]
7. y[2][0]
8. What position is x in?
9. What position is i in?
10. y[4][5]