

2-Dimensional Arrays

Arrays are data structures. This means you can store more than one item of data in a single variable.

For example:

```
testScores = [45, 76, 32, 98, 0]
```

But I'm likely to have more than one student with a set of scores...

2-Dimensional Arrays

I could set up an array for each student:

testScores1

testScores2

testScores3

etc...

But this gets tricky to use

2-Dimensional Arrays

A solution is to use an array of arrays:

```
testScores = [[45, 76, 32, 98, 0],  
[67, 34, 56, 23, 7], [0, 0, 0, 65,  
56]]
```

2-Dimensional Arrays

In a standard array I use the index to find or set the value of an individual element:

```
dog = ["Buffy", "Poodle", "Brown"]
```

`dog[0]` **gives me** "Buffy"

`dog[1] = "Rotweiler"` **sets the a new value**

`if dog[2] == "Brown":`

2-Dimensional Arrays

In a standard array I use the index to find or set the value of an individual element:

```
dog = ["Buffy", "Poodle", "Brown"]
```

`dog[0]` **gives me** "Buffy"

`dog[1] = "Rotweiler"` **sets the a new value**

`if dog[2] == "Brown":`

2-Dimensional Arrays

You use the indexes to get data from 2D arrays as well:

```
dogs = [ ["Buffy", "Poodle",  
"Brown"], ["Sally", "Pug", "White"],  
["Jeremy", "Spaniel", "Black"],  
["Fenton", "Labrador", "Brown"] ]
```

`dogs[2][0]` gives???

2-Dimensional Arrays

```
dogs = ["Buffy", "Poodle", "Brown",  
        "Sally", "Pug", "White", "Jeremy",  
        "Spaniel", "Black", "Fenton", "Labrador",  
        "Brown"]
```

`dogs[2][0]` gives "Jeremy"

`dogs[3]` gives ["Fenton", "Labrador", "Brown"]

`dogs[2][2]` ?

`dogs[0][0]` ?

`dogs[2][3]` ?

2-Dimensional Arrays

You can iterate over 2-D arrays in different ways:

```
for aDog in dogs:  
    print(aDog[0])  
    print(aDog[1])
```

Simple, easy to use method, usually works best

2-Dimensional Arrays

You can also iterate over 2-D arrays like this:

```
for i in range(0, len(dogs)):  
    for j in range(0, len(dogs[i]))  
        print(dogs[i][j])
```

Complex to code but can be better to use.

Guess which one the exam board prefers...

This works through my
dogs array in this order:

[0]	[0]
[0]	[1]
[0]	[2]
[1]	[0]
[1]	[1]
[1]	[2]
[2]	[0]
[2]	[1]
[2]	[2]
[3]	[0]

etc...

2-Dimensional Arrays

Think of the array as a table. The code works through cell by cell one row at a time. *i* is the row, *j* is the column

```
for i in range(0, len(dogs)):  
    for j in range(0, len(dogs[i]))  
        print(dogs[i][j])
```

row	column			
		0	1	2
	0	Buffy	Poodle	Brown
	1	Sally	Pug	White
	2	Jeremy	Spaniel	Black
	3	Fenton	Labrador	Brown

2-Dimensional Arrays

For example, count how many 0s there are - 0 means absent

```
testScores = [[45, 76, 32, 98, 0], [67, 34,  
56, 23, 7], [0, 0, 0, 65, 56]]
```

```
count = 0
```

```
for i in range(0, len(testScores)):  
    for j in range(0, len(testScores[i])):  
        if testScores[i][j] == 0:  
            count = count + 1
```

```
print("There are " + count + " absences")
```