Arrays and lists

When writing programs, it is useful to use arrays and lists as they simplify programs by storing related data under one name

What is an array?

A variable is a memory location that can store a value. It can be thought of as a box in which values are stored. The value held in the box can change, or vary. But each variable can only hold one item of data.

An array is a series of memory locations – or 'boxes' – each of which holds a single item of data, but with each box sharing the same name. All data in an array must be of the same data type.

For example, imagine that a score table in a game needs to record ten scores. One way to do this is to have a variable for each score:

score_0 score_1 score_2 score_3 score_4 score_5 score_6 score_7 score_8 score_9

This would work, but there is a better way. It is much simpler to keep all the related data under one name. We do this by using an array.

Instead of having ten variables, each holding a score, there could be one array that holds all the related data:

score(9)

By using this array, all 10 data items can be stored in one place.

Each 'box' in an array is referred to as an element.

It helps to think of an array as a row of cells, like the ones found in a table. Each cell represents an element:

The individual values, or array elements, are numbered 0 to 9 because computers start counting at 0.

Naming arrays

Arrays are named like variables. The number in brackets determines how many data items the array can hold. The array **score(9)** would allow ten data items to be stored.

Any facility that holds more than one item of data is known as a **data structure**. Therefore, an array is a data structure.

Using arrays

To store a data item in an array, the element that the data will be stored in needs to be referenced.

For example, in the score array **score(10)**, the following would store the score 3000 in the first element:

```
score[0] = 3000
```

The following would store the score 2500 in the second element:

```
score[1] = 2500
```

Computers start counting at 0, so the first element is [0], the second is [1], etc.
To access the data stored in an element, the element must be referred to by its number. For example:

```
print(score[0]) total = score[0] + score[1]
```

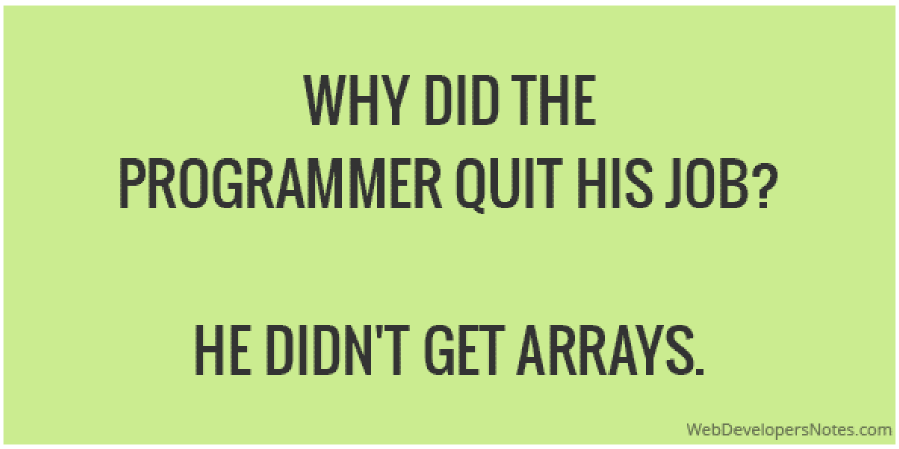
Why use arrays?

A variable holds a single item of data. There may be a situation where lots of variables are needed to hold similar and related data. In this situation, using an array can simplify a program by storing all related data under one name. This means that a program can be written to search through an array of data much more quickly than having to write a new line of code for every variable. This reduces the complexity and length of the program which makes it easier to find and debug errors.

Lists

Lists are data structures similar to arrays that allow data of more than one data type.

Some languages, such as BASIC and Java allow the use of arrays. Others, such as Python, only allow lists.


WHY DID THE
PROGRAMMER QUIT HIS JOB?

HE DIDN'T GET ARRAYS.

WebDevelopersNotes.com