Delphi Dimensional Arrays(2D)

A 2D array in Delphi is simply an array of arrays. It is a common way to represent data in a table or matrix format.

Item	Quantity	Price
Apple	3	R1.50
Banana	2	R0.75
Orange	1	R0.50

This table has three columns:

- "Item"
- "Quantity"
- "Price".

And it has four rows:

- "Item", "Quantity", "Price"
- "Apple", "3", R1.50
- "Banana", "2", "R0.75"
- "Orange", "1", "R0.50

Declare a 2D array

A 2D array can only be declared as one data type(String, Integer, Real), meaning that you cant have mixed data like strings and integers together.

Declare a 2D array

```
var
    arrTable: array[1..4,1..3] of string;
```

- The first `1..3` is the number of Rows
- The second `1..3` is the number of Columns.

Alternatively you could have used 0 based indexing:

```
var
    arrTable: array[0..3,0..2] of string;
```

Which is the same thing, just instead of starting from 1 we are now starting from 0. Both declarations say:

- you want an array of string data type
- `0..3` means four rows
- `0..2` means three columns

Declare a 2D array with values

```
arrTableValues: array [0 .. 3, 0 .. 2] of string = (('Item', 'Quantity',
      'Price'), ('Apple', '3', 'R1.50'), ('Banana', '2', 'R0.75'),
    ('Orange', '1', 'R0.50'));
```

Printing the data inside a 2D array

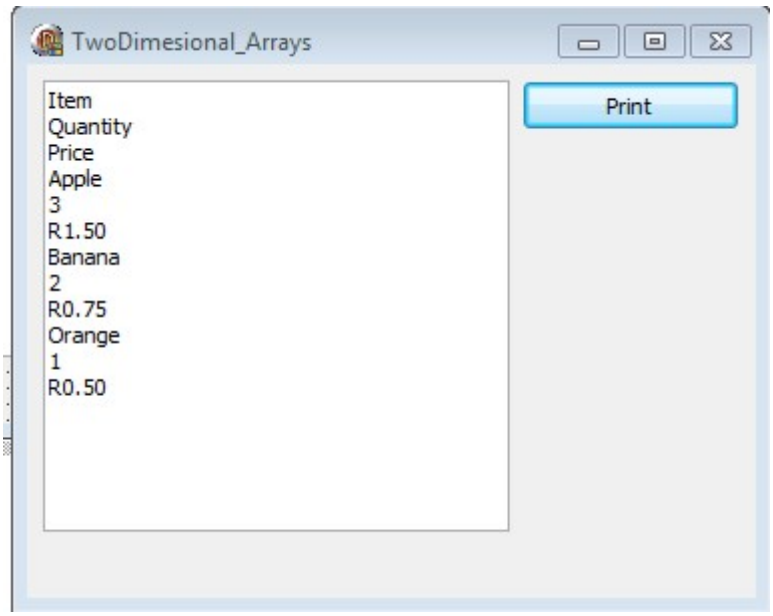
When working with 2D arrays, always remember this:

1. I will need two for loops.
    1. One for loop for my rows
    2. Second for loop for my columns

```
var
  row, col: integer;
begin
  for row := 0 to 3 do
  begin
    for col := 0 to 2 do
    begin

    end;
  end;

end;
```

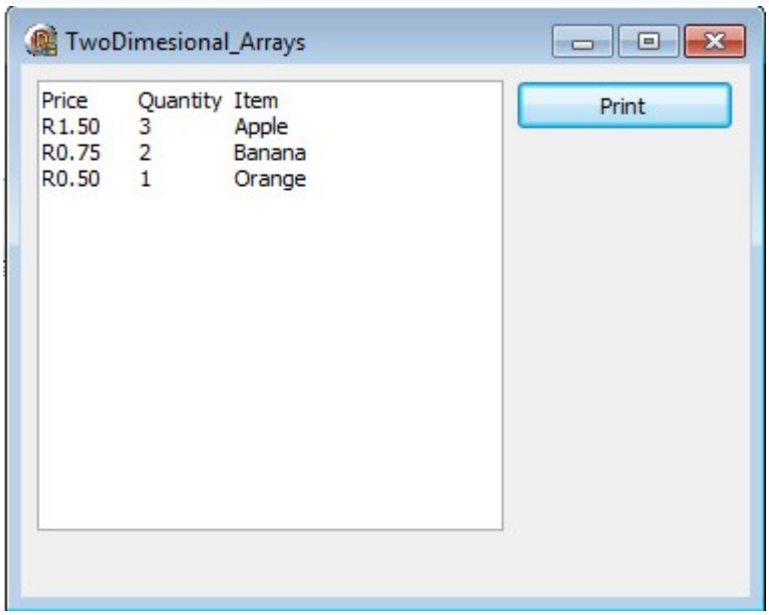
We can output our 2D array to o a richEdit:

```
var
  row, col: integer;
begin
  for row := 0 to 3 do
  begin
    for col := 0 to 2 do
    begin
      redOutput.Lines.Add(arrTableValues[row,col]);
    end;
  end;

end;
```



Printing with formatting

```
var
  row, col: integer;
  output: string;
begin
  for row := 0 to 3 do
  begin
    output := '';
    for col := 0 to 2 do
    begin
      output := arrTableValues[row,col] + #9 + output;
    end;
    redOutput.Lines.Add(output);
  end;

end;
```



Adding the Rows and Columns of a 2D Array

Consider this table:

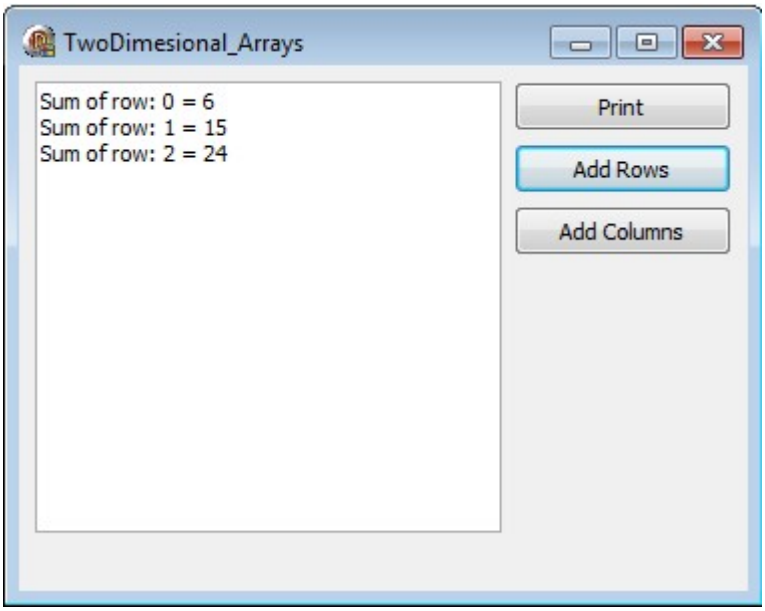
1	2	3
4	5	6
7	8	9

This table consists of integer values. There are 3 rows and 3 columns. Therefore the declaration for this 2D array will look like this:

```
arrInt: array [0 .. 2, 0 .. 2] of integer = ((1, 2, 3), (4, 5, 6), (7, 8, 9));
```

Adding the Rows

```
var
  row, col, rowSum: integer;
begin
  for row := 0 to 2 do
  begin
    rowSum := 0;
    for col := 0 to 2 do
    begin
      rowSum := arrInt[row, col] + rowSum;
    end;
    redOutput.Lines.Add('Sum of row: ' + inttostr(row) + ' = ' + inttostr(rowSum)
      );
  end;
end;
```



Adding the Columns

```
var
  row, col, colSum: integer;
begin
  for col := 0 to 2 do
  begin
    colSum := 0;
    for row := 0 to 2 do
    begin
      colSum := arrInt[row, col] + colSum;
    end;
    redOutput.Lines.Add('Sum of column: ' + inttostr(col) + ' = ' + inttostr
        (colSum));
  end;
end;
```