2D Array

2D array can be defined as an array of arrays. The 2D array is organized as matrices which can be represented as the collection of rows and columns.

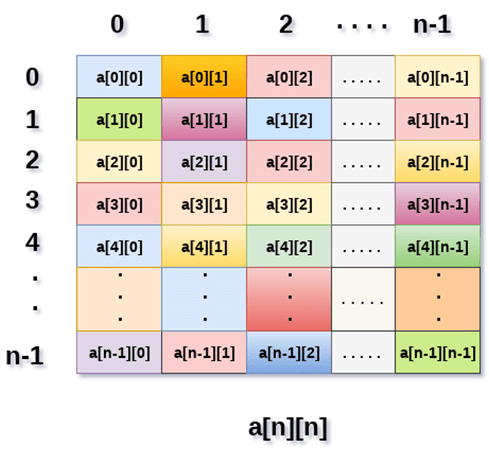
However, 2D arrays are created to implement a relational database look alike data structure. It provides ease of holding bulk of data at once which can be passed to any number of functions wherever required.

How to declare 2D Array

The syntax of declaring two dimensional array is very much similar to that of a one dimensional array, given as follows.

1. **int** arr[max\_rows][max\_columns];

however, It produces the data structure which looks like following.



Above image shows the two dimensional array, the elements are organized in the form of rows and columns. First element of the first row is represented by a[0][0] where the number shown in the first index is the number of that row while the number shown in the second index is the number of the column.

How do we access data in a 2D array

Due to the fact that the elements of 2D arrays can be random accessed. Similar to one dimensional arrays, we can access the individual cells in a 2D array by using the indices of the cells. There are two indices attached to a particular cell, one is its row number while the other is its column number.

However, we can store the value stored in any particular cell of a 2D array to some variable x by using the following syntax.

1. **int** x = a[i][j];

where i and j is the row and column number of the cell respectively.

We can assign each cell of a 2D array to 0 by using the following code:

**for** ( **int** i=0; i<n ;i++)

{

**for** (**int** j=0; j<n; j++)

    {

        a[i][j] = 0;

    }

}

Initializing 2D Arrays

We know that, when we declare and initialize one dimensional array in C programming simultaneously, we don't need to specify the size of the array. However this will not work with 2D arrays. We will have to define at least the second dimension of the array.

The syntax to declare and initialize the 2D array is given as follows.

1. **int** arr[2][2] = {0,1,2,3};

The number of elements that can be present in a 2D array will always be equal to (**number of rows \* number of columns**).

**Example :**

Python provides many ways to create 2-dimensional lists/arrays. However one must know the differences between these ways because they can create complications in code that can be very difficult to trace out. Lets start by looking at common ways of creating 1d array of size N initialized with 0s.

**Method 1a**

|  |
| --- |
| # First method to create a 1 D array  N = 5  arr = [0]\*N  print(arr) |

**Output:**

[0, 0, 0, 0, 0]

**Method 1b**

|  |
| --- |
| # Second method to create a 1 D array  N = 5  arr = [0 for i in range(N)]  print(arr) |

**Output:**

[0, 0, 0, 0, 0]

Extending the above we can define 2-dimensional arrays in the following ways

.  
**Method 2a**

|  |
| --- |
| # Using above first method to create a  # 2D array  rows, cols = (5, 5)  arr = [[0]\*cols]\*rows  print(arr) |

**Output:**

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

**Method 2b**

|  |
| --- |
| # Using above second method to create a  # 2D array  rows, cols = (5, 5)  arr = [[0 for i in range(cols)] for j in range(rows)]  print(arr) |

**Output:**

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

**Method 2c**

|  |
| --- |
| # Using above second method to create a  # 2D array  rows, cols = (5, 5)  arr=[]  for i in range(cols):      col = []      for j in range(rows):          col.append(0)      arr.append(col)  print(arr) |

**Output:**

[[0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0], [0, 0, 0, 0, 0]]

Both the ways give seemingly same output as of now. Lets change one of the elements in the array of method 2a and method 2b.

|  |
| --- |
| # Python 3 program to demonstrate working  # of method 1 and method 2.    rows, cols = (5, 5)    # method 2a  arr = [[0]\*cols]\*rows    # lets change the first element of the  # first row to 1 and print the array  arr[0][0] = 1    for row in arr:      print(row)  # outputs the following  #[1, 0, 0, 0, 0]  #[1, 0, 0, 0, 0]  #[1, 0, 0, 0, 0]  #[1, 0, 0, 0, 0]  #[1, 0, 0, 0, 0]    # method 2b  arr = [[0 for i in range(cols)] for j in range(rows)]    # again in this new array lets change  # the first element of the first row  # to 1 and print the array  arr[0][0] = 1  for row in arr:      print(row)    # outputs the following as expected  #[1, 0, 0, 0, 0]  #[0, 0, 0, 0, 0]  #[0, 0, 0, 0, 0]  #[0, 0, 0, 0, 0]  #[0, 0, 0, 0, 0] |

**Output:**

[1, 0, 0, 0, 0]

[1, 0, 0, 0, 0]

[1, 0, 0, 0, 0]

[1, 0, 0, 0, 0]

[1, 0, 0, 0, 0]

[1, 0, 0, 0, 0]

[0, 0, 0, 0, 0]

[0, 0, 0, 0, 0]

[0, 0, 0, 0, 0]

[0, 0, 0, 0, 0]