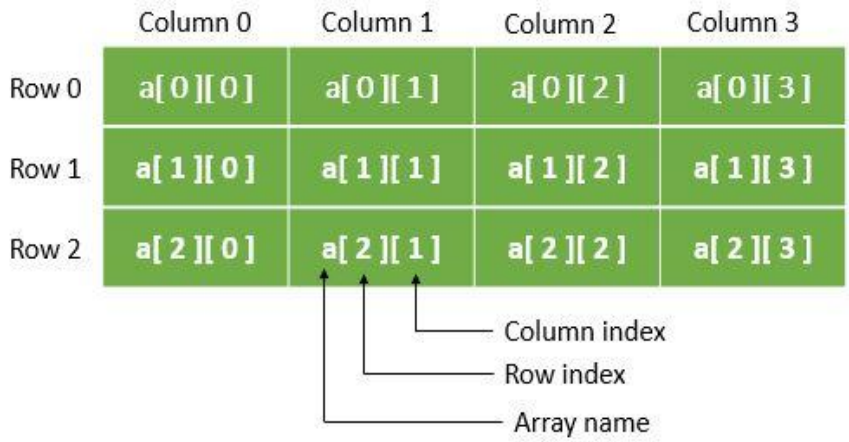

Two dimensional array :-

Initialization

```
int arr[4][2] =
 {
        {1234, 56},
        {1212, 33},
        {1434, 80},
        {1312, 78}
} ;
```

or

```
int arr[4][2] = {1234, 56, 1212, 33, 1434, 80, 1312, 78};
```

Printing a 2D Array in C

We are done initializing a 2D array, now without actually printing the same, we cannot confirm that it was done correctly.

Also, in many cases, we may need to print a resultant 2D array after performing some operations on it. So how do we do that?

The code below shows us how we can do that.

```
#include<stdio.h>

int main( )
{
```

```
        int arr[4][2] = {
                { 10, 11 },
                { 20, 21 },
                { 30, 31 },
                { 40, 41 }
                } ;

        int i,j;

        printf("Printing a 2D Array:\n");
        for(i=0;i<4;i++)
        {
                for(j=0;j<2;j++)
                {
                        printf("%d \t",arr[i][j]);
                }
                printf(“ \n");
        }
Return 0;
}
```

Taking 2D Array Elements As User Input

Previously, we saw how we can initialize a 2D array with pre-defined values. But we can also make it a user input too. Let us see how

```
#include<stdio.h>

int main( )
{
    int  s[2][2];
    int  i, j;
    printf("\n2D Array Input:\n");
          for(i=0;i<2;i++)
          {
                for(j=0;j<2;j++)
                {

                        scanf("%d",&s[i][j]);
                }
          }

          printf("The 2-D Array is:\n");
    for(i=0;i<2;i++)
          {
```

```
                for(j=0;j<2;j++)
                {
                        printf("%d \t",s[i][j]);
                }
                printf(“\n");
        }
return 0;
}
```