Google Developer Student Clubs

Intro To Array In c

Taher Mohamed

Array

- Array is a group of data that holds fixed number of value all of them are the same type.

- Syntax

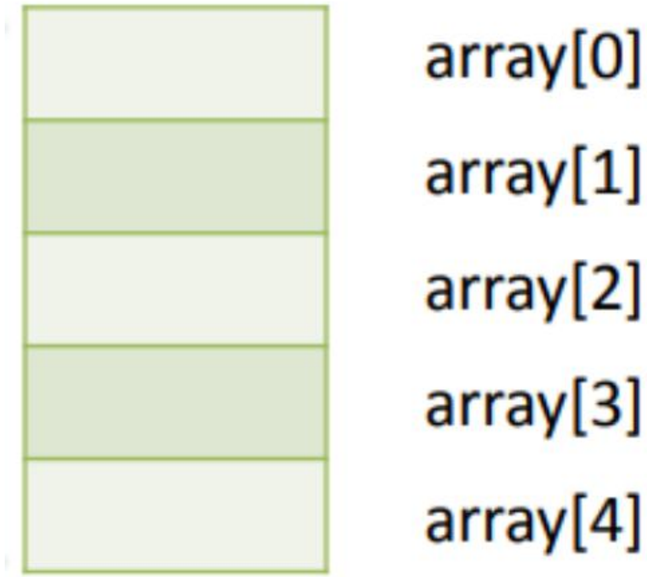
```
Array_Type  Array_Name[Length];

Array_Type  like  (int , char , double .....)
```

- Example

  int array [5] ;

  This line creates an array of int of size 5.



Notes:

1- The array index always starts from 0, it means that this array has elements from element 0 till element 4.

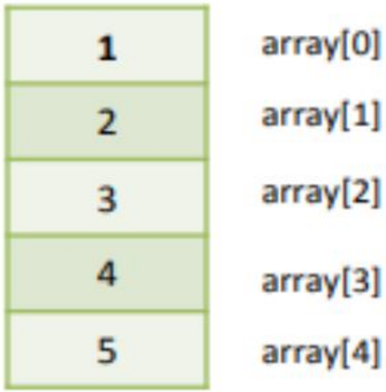
2- Array length must be constant value, can not be variable

Array Initialization

- Array can be initialized at the time of definition. To initialize the arry use the following syntax:

```
Array_Type  Array_Name [Length] = { values separated by comma };
```

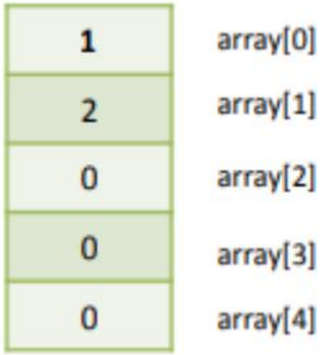
```
int array [5]={1,2,3,4,5};
```



Special Cases

1- Initializing the array with values less than its length, the remaining elements will be initialized with 0.

```
array [5] = { 1 , 2 } ;
```



2- Initializing the array with values more than its length, it would gives a compilation error.

```
array [4] = { 1 , 2 , 3 , 4 , 5 } ;
```
(compilation error)

Accessing Array Elements

- All elements of the array can be accessed at the same statement only at initialization. After initialization the array can be accessed only element by element.

- Syntax

```
Array_Name [Element_Index] ;

array [ 1 ] = 3 ;

printf ( " Element 1 = %d " , array [ 1 ] ) ;
```

Note:

1- Again, array index starts from 0.

2- You can use a variable to indicate for the element index,
for ex:

```
array [ i ] = 10;
```

where i is variable equals to the desired index number.

LAB1

Write a C code that ask the user to enter 10 values and save them in an array using a for loop. Then print the values entered by the user in reverse order using another for loop.

Expected Output

```
Please Enter number 0: 5
Please Enter number 1: 6
Please Enter number 2: 7
Please Enter number 3: 8
Please Enter number 4: 9
Please Enter number 5: 10
Please Enter number 6: 11
Please Enter number 7: 12
Please Enter number 8: 13
Please Enter number 9: 14
The values in reversed order
14
13
12
11
10
9
8
7
6
5
```

```c
#include <stdio.h>

void main(void)
{
    int array[10];
    int i;

    for (i=0;i<10;i++)
    {
        printf("Please Enter number %d: ",i);
        scanf("%d",&array[i]);
    }

    printf("The values in reversed order\n");

    for ( i =9; i>=0; i--)
    {
        printf("%d\n",array[i]);
    }
}
```

LAB2

Write a C code that ask the user to enter 10 values and save them in an array using a for loop. Then print the summation and the average of the values entered.

Expected Output

```
Please Enter number 0: 10
Please Enter number 1: 20
Please Enter number 2: 30
Please Enter number 3: 40
Please Enter number 4: 50
Please Enter number 5: 60
Please Enter number 6: 70
Please Enter number 7: 80
Please Enter number 8: 90
Please Enter number 9: 100
Sum of array elements = 550
Average of array elements = 55
```


```c
#include <stdio.h>

void main(void)
{
    int array[10];
    int i;
    int sum =0;
    int avg;

    for (i=0;i<10;i++)
    {
        printf("Please Enter number %d: ",i);
        scanf("%d",&array[i]);
        sum += array[i];
    }

    avg = sum/10;

    printf("Sum of array elements = %d\n",sum);
    printf("Average of array elements = %d",avg);
}
```

LAB3

Write a C code that ask the user to enter 10 values and save them in an array using a for loop. Then print the minimum and the maximum of the values.

Google Developer Student Clubs

```c
#include<stdio.h>

void main (void)
{
    int i;
    int arr[10];
    int max,min;

    /* Scan the values loop */
    for (i=0;i<10;i++)
    {
        printf("Please Enter number %d:  ",i);
        scanf ("%d",&arr[i]);
    }

    max=arr[0];
    min=arr[0];

    for(i=0;i<10;i++)
    {
        if(arr[i] > max)
        {
            max = arr[i];
        }

        if(arr[i]<min)
        {
            min=arr[i];
        }
    }

    printf("the maximum = %d\n",max);
    printf("the minimum = %d",  min);
}
```

Assignment1

Write C code that manage a small school. The school has 3 classes each class contains 10 students. Define three arrays for the three classes each one with a length of 10. Save a random numbers in all array elements to indicate the students grade. The program will calculate and display the following statistics:

1- Number of passed students
2- Number of Failed students
3- Highest grade
4- Lowest grade
5- Average grade Knowing that the total grade is from 100 and the minimum passing grade is 50

Google Developer Student Clubs

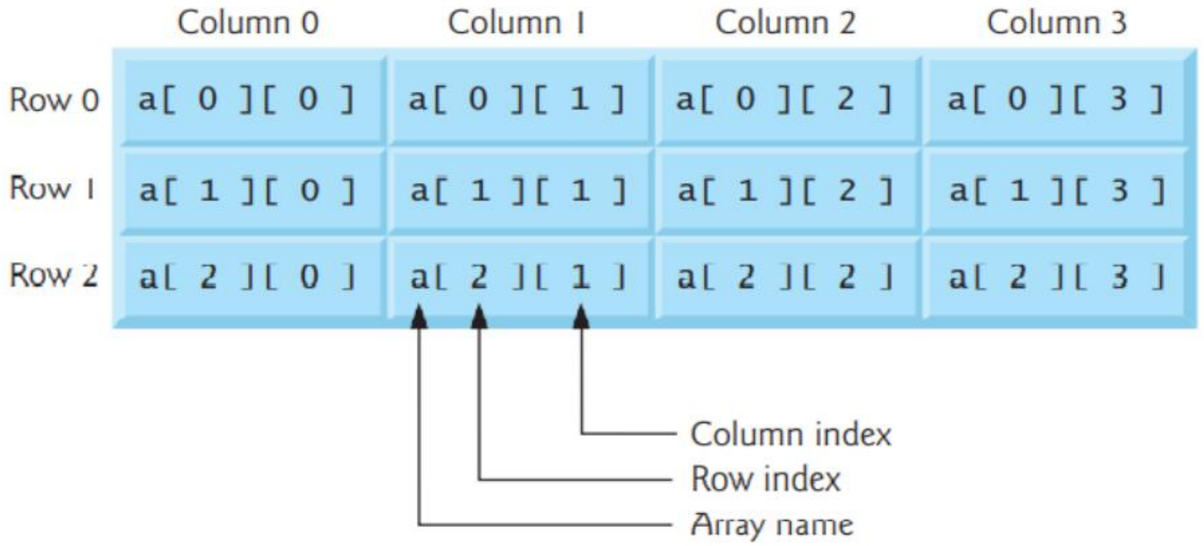
Multidimensional Arrays

Syntax

array_type array_name [ROW][Column];

Example:

int b [3][4] ;



Accessing Array Elements:

b [0 , 0] = 0;

Note:

1-A multidimensional array can be initialized when it's defined

```
int b[ 2 ][ 2 ] = { { 1, 2 }, { 3, 4 } };
```

2-If there are not enough initializers for a given row, the remaining elements of that row are initialized to 0.

```
int b[ 2 ][ 2 ] = { { 1 }, { 3, 4 } };
```

b[0][1] to 0

3-we can write a multidimensional array without row index but must write column index

```
int a [ ][ 3 ] ;
```


- For example:

Write C code that print three multidimensional array but use function

Output
Values in array1 by row are:
1 2 3 4 5 6
Values in array2 by row are:
1 2 3 4 5 0
Values in array3 by row are:
1 2 0 4 0 0

```c
// Initializing multidimensional arrays.
#include <stdio.h>

void printArray( int a[][ 3 ] ); // function prototype

// function main begins program execution
int main( void )
{
   int array1[ 2 ][ 3 ] = { { 1, 2, 3 }, { 4, 5, 6 } };
   int array2[ 2 ][ 3 ] = { 1, 2, 3, 4, 5 };
   int array3[ 2 ][ 3 ] = { { 1, 2 }, { 4 } };

   puts( "Values in array1 by row are:" );
   printArray( array1 );

   puts( "Values in array2 by row are:" );
   printArray( array2 );

   puts( "Values in array3 by row are:" );
   printArray( array3 );
} // end main
// function to output array with two rows and three columns
void printArray( int a[][ 3 ] )
{
  size_t i; // row counter
  size_t j; // column counter

  // loop through rows
  for ( i = 0; i <= 1; ++i ) {

  // output column values
  for ( j = 0; j <= 2; ++j ) {
  printf( "%d ", a[ i ][ j ] );
 } // end inner for

 printf( "\n" ); // start new line of output
 } // end outer for
} // end function printArray
```

String

string is a sequence of characters terminated with a null character \0.

For example :

char c [ ] = "c string“ ;

declare strings:

char s [5] ;

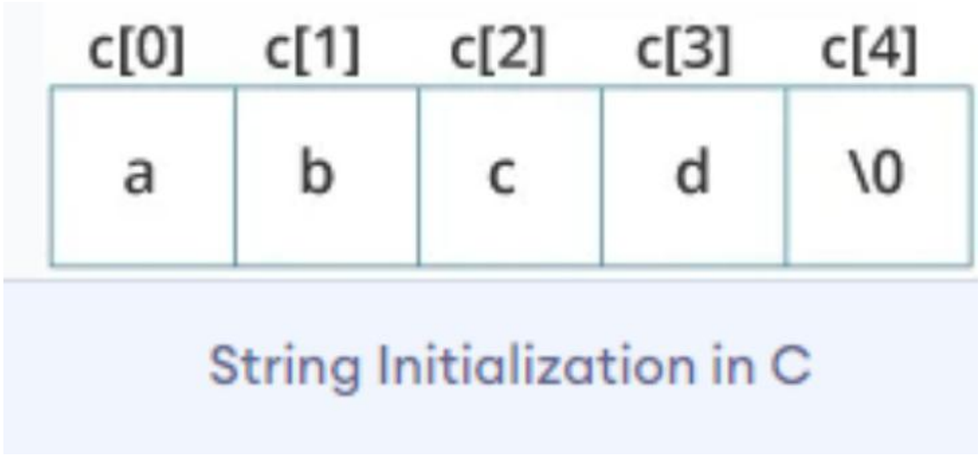
initialize strings in a number of ways.

```
char c[ ] = "abcd";

char c[50] = "abcd";

char c[] = {'a', 'b', 'c', 'd', '\0'};

char c[5] = {'a', 'b', 'c', 'd', '\0'};
```



String Initialization in C

Example:

1-Read String from the user

```c
#include <stdio.h>
int main()
{
    char name[20];
    printf("Enter name: ");
    scanf("%s", name);
    printf("Your name is %s.", name);
    return 0;
}
```

Output

```
Enter name: Dennis Ritchie
Your name is Dennis.
```