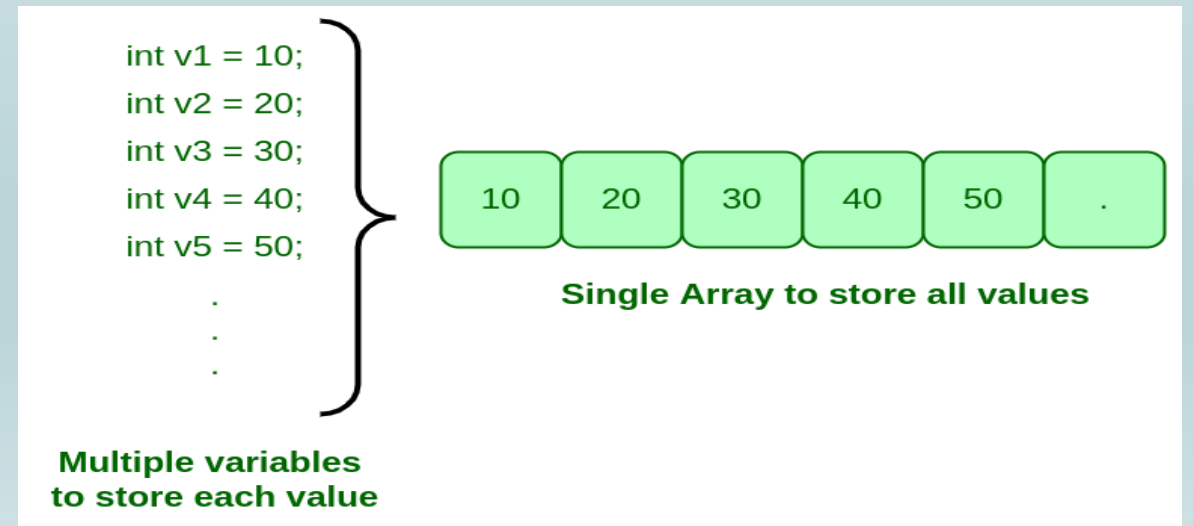
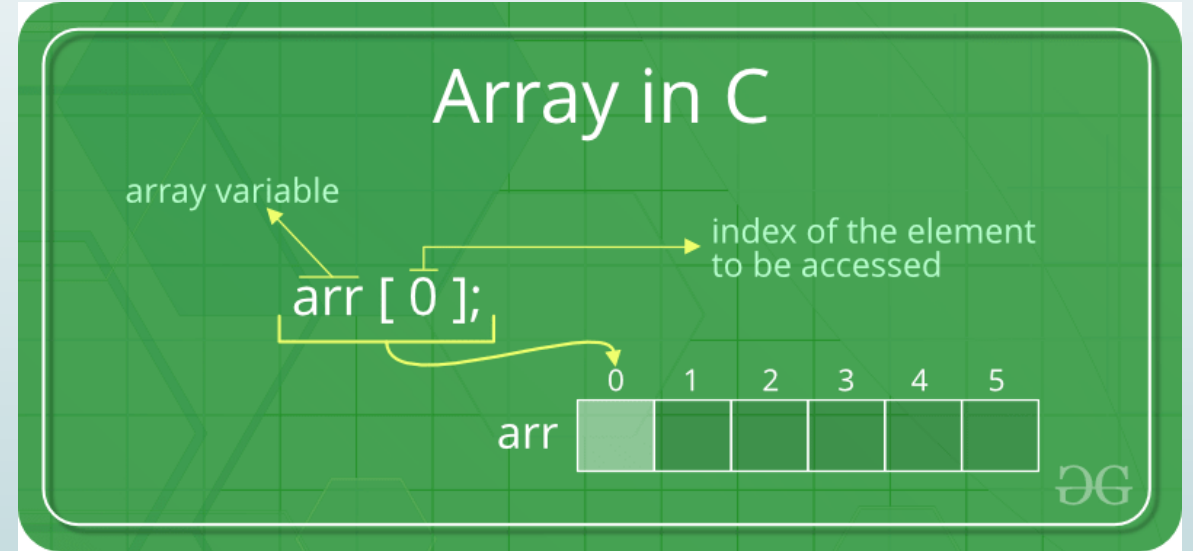


DİZİLER

Hazırlayan : Muhammed Esad Goncaloğlu

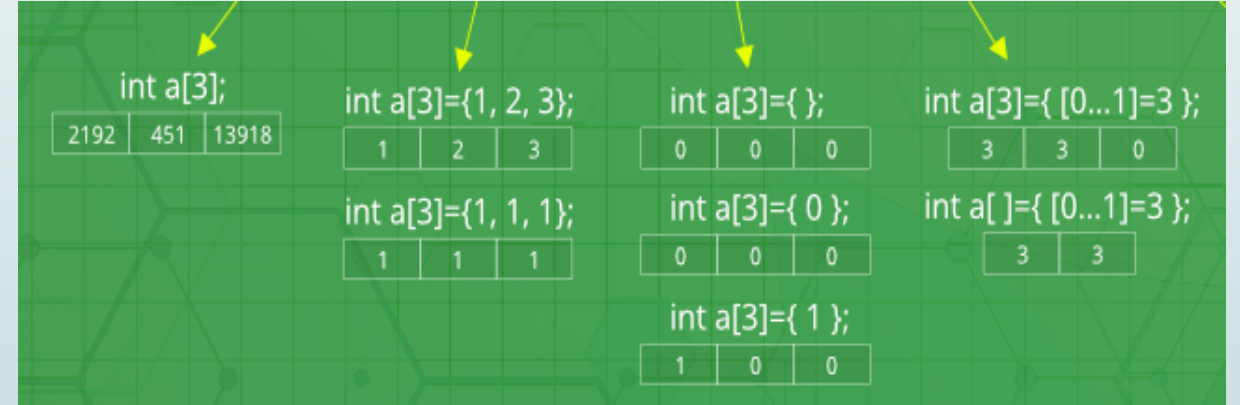
DİZİLER

- Diziler, bitişik bellek konumlarında depolanan benzer türde veri öğelerinin toplanması olarak tanımlanırlar.
- Diziler, C programlama dilinde int, char, double, float gibi ilkel tipteki verileri saklayabilen türetilmiş veri tipidir.
- Ayrıca işaretçiler, yapı gibi türetilmiş veri tiplerinin toplanmasını da saklama yeteneğine sahiptir.
- Diziler, her bir veri elemanına indeks numarası kullanılarak rastgele erişilebilen veri yapılarıdır.



DİZİLERDE DEĞER ATAMA

- Dizi öğeleri her zaman sıfırdan (0) itibaren sayılır.
- Dizilerin boyutları bir sayı ile de belirtilebilir ya da tamsayı bir değişken ile de belirtilebilir.
- Dizilere değerler dizi tanımlanırken de tanımlanabilir daha sonradan indekslere tanımlanabilir.
- Diziler başlangıçta belirtildiği kadar değer alabilirler. Diğer girilen değerler aşırı yüklenmeden dolayı kaybedilebilir.
- Eğer dizilere eleman kadar değer eklenmek isteniyorsa boyut kısmı boş bırakılarak değerler tanımlanır.



```
int arr1[10];
```

```
int n = 10;
```

```
int arr2[n];
```

```
int arr[5];  
arr[0] = 5;  
arr[2] = -10;  
arr[3 / 2] = 2;  
arr[3] = arr[0];
```

DİZİLERDE DEĞER YAZDIRMA

- Dizilerde değerleri döngülerle yazdırmak daha kolay ve faydalıdır.
- Döngüler sayesinde dizileri yazdırabilir dizilere değer atayabilir dizilerle çeşitli operatörler sayesinde işlemler gerçekleştirebiliriz.
- printf() fonksiyonunda dizinin indekslerindeki değerlerin iç içe girmemesi için '\n' kaçış komutunu kullandık.
- Böylece değerlerimizi alt alta düzgün bir şekilde yazdırmış olduk.

```
int arr[6] = {11, 12, 13, 14, 15, 16};  
// Way 1  
for(int i = 0; i < 6; i++){  
    printf("%d\n", arr[i]);  
}  
return 0;
```

```
int arr[6];  
for(int i = 0; i < 6; i++){  
    printf("Please enter %d. element of the array : ",i);  
    scanf("%d",&arr[i]);  
}  
for(int i = 0; i < 6; i++){  
    printf("%d\n", arr[i]);  
}  
return 0;
```

sizeof() FONKSİYONU

- sizeof() fonksiyonu, bize içine verdiğimiz veri tipinin bellekte ne kadar bit alan kapladığını döndürür.
- Oldukça basit bir fonksiyon olmasına karşın oldukça kullanışlıdır.
- Veri tiplerinin bellekte ne kadar alan kapladığından dizilerdeki eleman sayısını hesaplamaya kadar geniş bir kullanım alanı vardır.

```
int arr[6];  
for(int i = 0; i < sizeof(arr)/sizeof(int); i++){  
    printf("Please enter %d. element of the array : ",i);  
    scanf("%d",&arr[i]);  
}  
for(int i = 0; i < sizeof(arr)/sizeof(int); i++){  
    printf("%d\n", arr[i]);  
}  
return 0;
```

```
printf("Size of int data type:%d\n",sizeof(int));  
printf("Size of char data type:%d\n",sizeof(char));  
printf("Size of float data type:%d\n",sizeof(float));  
printf("Size of double data type:%d\n",sizeof(double));  
printf("Size of int data type:%d\n",sizeof(a));  
printf("Size of an int array:%d\n",sizeof(b));  
return 0;
```


©tutorialgateway.org

C:\Users\Personal\Documents\SizeOf Exmple in C.exe

```
Size of int data type:4  
Size of char data type:1  
Size of float data type:4  
Size of double data type:8  
Size of int data type:4  
Size of an int array:80
```

ÇOK BOYUTLU DİZİLER

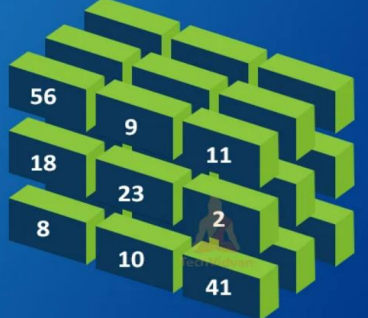
- İki veya daha fazla boyuttan oluşan dizilere çok boyutlu diziler denir.
- Genellikle büyük verilerin depolanmasında kullanılır.
- İki boyutlu diziler aynı zamanda matris olarak da adlandırılır. Tıpkı Excel'deki gibi satır sütun mantığı ile veriler depolanır.
- Üç boyuttan daha büyük diziler düşünülmesi oldukça zordur.



Multidimensional Arrays in C

	column 0	column 1	column 2
Row 0	a[0][0]	a[0][1]	a[0][2]
Row 1	a[1][0]	a[1][1]	a[1][2]
Row 2	a[2][0]	a[2][1]	a[2][2]
Row 3	a[3][0]	a[3][1]	a[3][2]
Row 4	a[4][0]	a[4][1]	a[4][2]

2-D Array



3-D Array

```
int arr[3][3];
for(int i = 0; i < 3; i++){
    for(int j = 0; j < 3; j++){
        printf("Please enter [%d,%d] of the array : ",i,j);
        scanf("%d",&arr[i][j]);
    }
}

for(int i = 0; i < 3; i++){
    for(int j = 0; j < 3; j++){
        printf("%d\t", arr[i][j]);
    }
    printf("\n");
}
```

KAYNAKÇA

- <https://www.javatpoint.com/c-array>
- <https://www.geeksforgeeks.org/arrays-in-c-cpp/>
- <https://www.tutorialgateway.org/c-sizeof-operator/>
- <https://www.programiz.com/c-programming/c-multi-dimensional-arrays>
- <https://techvidvan.com/tutorials/multidimensional-arrays-in-c/>