

C Dilindeki Tüm Anahtar Kelimelerin Listesi

Bu döküman , C programlamadaki tüm 32 anahtar kelime hakkında kısa bir bilgi sağlar.

C'deki Tüm Anahtar Kelimelerin Açıklaması :

- **Auto**

Auto anahtar sözcüğü otomatik değişkenleri bildirir.
Örneğin:

```
auto int var1;
```

Bu ifade, var1'in auto saklama sınıfı ve int türünde bir değişken olduğunu gösterir.

İşlev gövdelerinde bildirilen değişkenler varsayılan olarak otomatiktir. Bir işlev her çalıştırıldığında yeniden oluşturulurlar.

Otomatik değişkenler bir işleve yerel olduğundan, aynı zamanda yerel değişkenler olarak da adlandırılırlar.

- **break and continue**

Break ifadesi, programın en içten (while, do, for veya switch) en dışa atlamasını sağlar.

continue ifadesi, döngü içindeki belirli ifadeleri atlar.

```
for (i=1;i<=10;++i)
{
    if (i==3)
        continue;
    if (i==7)
        break;
    printf("%d ",i);
}
```

Çıktı :

1 2 4 5 6

İ, 3'e eşit olduğunda, *continue* ifadesi yürürlüğe girer ve 3 değerini atlar. İ, 7'ye eşit olduğunda, *break* ifadesi yürürlüğe girer ve for döngüsünü sonlandırır.

▪ switch, case ve default

Switch ve case deyimi birçok blok arasında “bir deyim bloğu” çalıştırılması gerektiğinde kullanılır. Örneğin:

```
switch(expression)
{
case '1':      //1 olduğunda çalıştırılacak ifadeler
```

```
break;
case '5':      //5 olduğunda çalıştırılacak ifadeler
break;
default:      //varsayılan olduğunda çalıştırılacak bazı
ifadeler;
}
```

▪ char

Char anahtar sözcüğü bir karakter değişkeni bildirir. Örneğin:

```
char alfabe;
```

Burada, alfabe bir karakter tipi değişkenidir.

▪ const

Bir tanımlayıcı, const anahtar sözcüğü kullanılarak sabit olarak bildirilebilir.

```
const int a = 5;
```

▪ do...while

```
int i;
do
{
    print("%d ",i);
    i++;
}
```

```
while (i<10)
```

▪ double ve float

İkili ve değişkenli anahtar kelimeler, tür değişkenlerini bildirmek için kullanılır. Örneğin:

```
float number;  
double longNumber;
```

Burada sayı, tek duyarlıklılı ondalıklı sayı değişkeni iken, longNumber, çift duyarlıklılı ondalıklı sayı değişkenidir.

▪ if ve else

C programlamada, karar vermek if ve else kullanılıyor.

```
if (i == 1)    printf("i is 1.") else    printf("i is not 1.")
```

Eğer i değeri 1'den farklı ise, çıktısı şöyle olur:

```
i is not 1
```

▪ enum

Numaralandırma tipleri, C programlamasında enum anahtar kelimesi kullanılarak bildirilmiştir. Örneğin:

```
enum suit
{
    hearts;
    spades;
    clubs;
    diamonds;
};
```

▪ **extern**

Extern anahtar sözcüğü, bir değişkenin veya işlevin, bildirilen dosyanın dışında harici bir bağlantıya sahip olduğunu bildirir.

▪ **for**

C programlamasında üç tür döngü vardır. For döngüsü, işlemlerin tekrar sayısının önceden belli olduğu durumlarda kullanılır. Örneğin:

```
for (i=0; i< 9;++i)
{
    printf("%d ",i);
}
```

Çıktısı :

0 1 2 3 4 5 6 7 8

▪ **goto**

Goto anahtar sözcüğü, bir işlem içindeki etiketli bir ifadeye koşulsuz atlama için kullanılır. Örneğin:

```
for(i=1; i<5; ++i)
{
    if (i==10)
        goto error;
}
printf("i is not 10");
error:
    printf("Error, count cannot be 10.");
```

Çıktısı :

Error, count cannot be 10.

▪ **int**

Int anahtar sözcüğü tamsayı tipi değişkeni bildirir. Örneğin:

```
int count;
```

Burada, count bir tamsayı değişkenidir.

▪ **short, long, signed ve unsigned**

Bunlar, yeni bir tür vermek için temel veri türünün anlamını değiştiren tür değiştiricileridir.

```
short int smallInteger;  
long int bigInteger;  
signed int normalInteger;  
unsigned int positiveInteger;
```

Data types	Range
short int	-32768 to 32767
long int	-2147483648 to 214743648
signed int	-32768 to 32767
unsigned int	0 to 65535

▪ **return**

Return anahtar sözcüğü işlevi sonlandırır ve değeri döndürür.

```
int func()  
{  
    int b = 5;  
    return b;  
}
```

▪ **sizeof**

Sizeof anahtar sözcüğü verinin boyutunu değerlendirir.

```
#include <stdio.h>  
int main()  
{  
    printf("%u bytes.",sizeof(char));  
}
```

Çıktısı :

1 bytes.

- **register**

Register anahtar sözcüğü, normal değişkenlerden çok daha hızlı olan register değişkenleri oluşturur.

```
register int var1;
```

- **static**

Statik anahtar kelime bir statik değişken oluşturur. Statik değişkenlerin değeri programın sonuna kadar devam eder. Örneğin:

```
static int var;
```

- **struct**

Struct anahtar sözcüğü bir yapı bildirmek için kullanılır. Bir yapı, farklı türdeki değişkenleri tek bir ad altında tutabilir.

```
struct student{  
    char name[80];  
    float marks;
```

```
    int age;  
}s1, s2;
```

▪ typedef

Typedef anahtar sözcüğü bir türü açıkça tanımlayıcıyla ilişkilendirmek için kullanılır.

```
typedef float kg;  
kg bear, tiger;
```

▪ union

Bir Union, farklı değişken türlerini tek bir ad altında gruplamak için kullanılır.

```
union student  
{  
    char name[80];  
    float marks;  
    int age;  
}
```

▪ void

Void anahtar sözcüğü, bir işlevin herhangi bir değer döndürmediğini gösterir.

```
void testFunction(int a)  
{
```

```
.....  
}
```

Burada, `testFunction ()` fonksiyonu bir değer döndüremiyor çünkü dönüş tipi geçersiz.

▪ **volatile**

Volatile , geçici nesneler oluşturmak için kullanılır. Bir geçici nesne, donanım tarafından belirsiz bir şekilde değiştirilebilir.

```
const volatile number
```

Burada sayı geçici bir nesnedir.

Sayı sabit bir değişken olduğundan, program onu değiştiremez. Ancak, donanım geçici bir nesne olduğundan onu değiştirebilir.