

# Bölüm 2
# C Dilinin Temelleri

- Genel Kavramlar
- Yazım ve Noktalama Kuralları
- C Kütüphaneleri
- C Dilindeki Sözcükler
- Değer Sabitleri
- Veri Tipleri
- Değişkenler

## Ana Fonksiyon: main( )

```
int main(void)
{
    program_komutları
    return(0);
}
```

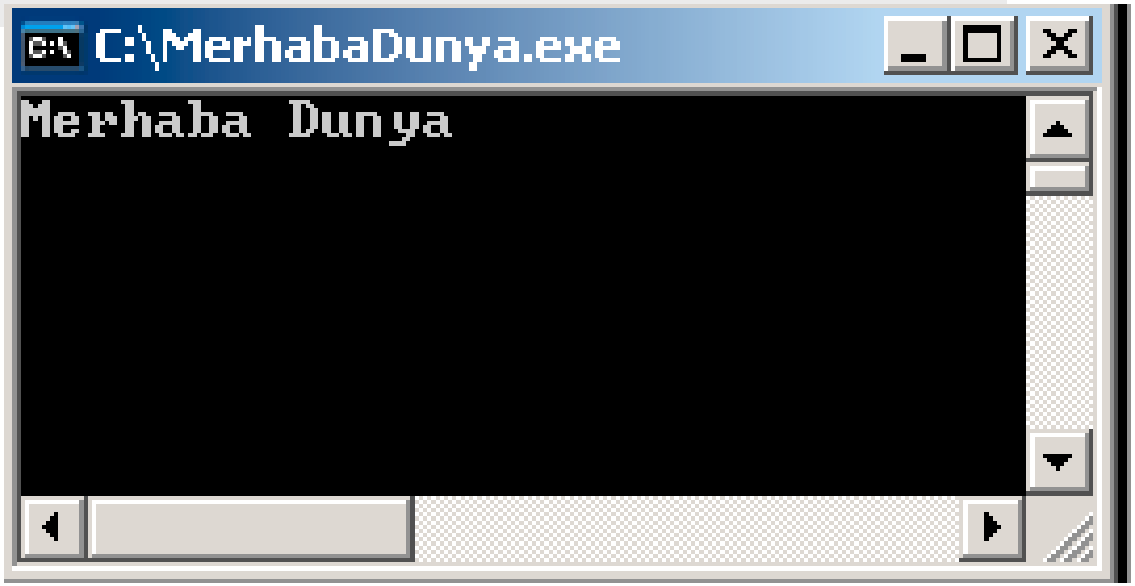
# Genel Kavramlar

**`return( ):`** Fonksiyonun çağrıldığı yere geri dönmesini sağlar. main( ) fonksiyonu içinde kullanılan return() komutu programın sonlandırılarak işletim sistemine geri dönülmesini sağlar.

## Örnek Program

```
#include <stdio.h>
int main (void)
{    printf("Merhaba Dunya");
     return(0);
}
```

**Çıktı:**



# Genel Kavramlar

`printf()`: çıktı fonksiyonu ekrana çeşitli çıktıların yazılması

```
printf("format_dizgisi");
```

Bu fonksiyonun programımız tarafından kullanılması için `<stdio.h>` kütüphanesinin program başında tanımlanması gerekir.

```
#include <stdio.h>
```

## Örnek Program

Ekranda “Bu benim ilk programim.” cümlesini görüntüleyecek programı yazınız.

```
#include <stdio.h>
int main(void)
{
    printf(“Bu benim ilk programim. “);
    return(0);
}
```

## Çıktı:

```
Bu benim ilk programim.
```

# Yazım ve Noktalama Kuralları

**Noktalı Virgül:** Her komutun bittiği yer ‘;’ işareti ile belirtilir.

**Ayraç İşaretleri:** C dili bloklardan oluşur. Bloklar ‘{‘ ve ‘}’ ayraç işaretleri içinde belirtilen komutların bir araya gelmesi ile oluşur.

**Açıklama Satırları:** Program içindeki açıklamanın başlangıç ve bitiş noktaları “/*” ve “*/” işaretleri ile belirlenmelidir.

```
#include <stdio.h>
int main(void)
{
printf(“Ilk program“);
return(0);
}
```

# Yazım ve Noktalama Kuralları

```
include <stdio.h>
int main (void)
{
/*Bu program bilgisayarinizin ekranina
Merhaba Dunya kelimelerini yazar.*/
     printf("Merhaba Dunya");
     return(0);
/*Burasi programin sonudur*/
}
```

# Yazım ve Noktalama Kuralları

- Bir bloğu oluşturan tüm komutlar ve açıklama satırları ayraç işaretlerine göre daha içeride yazılır.
- Böylece blokların birbirinden ayrılması ve yazılan program kodunun daha anlaşılır bir hale gelmesi sağlanır.

```
include <stdio.h>
int main (void)
{
/*Bu program ekrana
Merhaba kelimesini
yazar.*/
    printf("Merhaba");
    return(0);
/*Burasi programin
sonudur*/
}
```

# C Kütüphaneleri

- Yapılan işin özelliğine göre programa dahil edilir.
- Kodu daha önce yazılmış, programcının kullanımına hazır fonksiyonları içerir.
- Bu fonksiyonların her defasında tekrar tekrar yazılması engellenmiş olur ve daha standart bir yapıda yaygın olarak kullanımı sağlanır.

| | |
|---|---|
| Standart girdi/çıktı: | <stdio.h> |
| Matematiksel işlemler: | <math.h> |
| Dizgiler: | <string.h> |

# C Dilindeki Sözcükler

```
/*Bu program dairenin alanini bulur*/
#include <stdio.h>
#define PI 3.14
int main(void)
{
  double varicap, alan;
  printf("Yaricapi giriniz:");
  scanf("%lf", &varicap);
  alan = PI * varicap * varicap;
  printf(" Alan = %f", alan);
  return(0);
}
```

açıklama

kütüphane

isim sabiti

ana fonksiyon

tanımlama komutu

Özel amaçlı sözcükler

Tanıtıcı isimleri (değişken)

Tanıtıcı isimleri (fonksiyon)

## Özel Amaçlı Sözcükler

| | | | |
|---|---|---|---|
| auto | double | if | static |
| break | else | int | struct |
| case | entry | long | switch |
| char | extern | register | typedef |
| const | enum | return | union |
| continue | float | sizeof | unsigned |
| default | for | signed | void |
| do | goto | short | while |

# C Dilindeki Sözcükler

**Tanıtıcılar:** Özel amaçlı sözcüklerin dışında yapılan tanımlamalardır. Tanıtıcı isimleri

- içinde harf (a ..z, A.. Z), sayı (0..9) veya alt çizgi (underscore '_') bulunabilir.
- bir harf ya da alt çizgi '_' işareti ile başlamalıdır.
- içinde özel karakterler ( #, $, &, ö, ş " v.b. ) bulunamaz.
- C dilindeki özel amaçlı sözcüklerden biri olamaz.
- büyük-küçük harfler farklı (case-sensitive) algılanır.

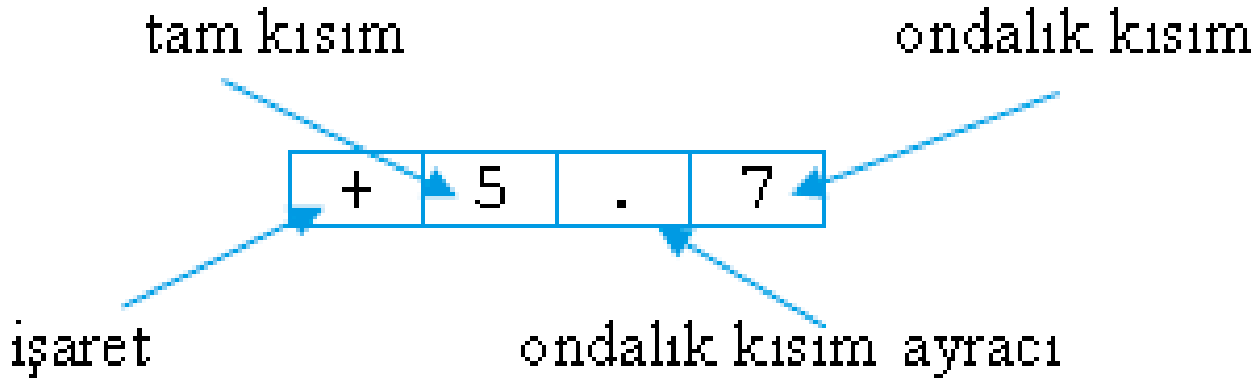
**Örnek:** ogrenci_yasi _06_yili_ucreti ucret Short

# Değer Sabitleri

**Tamsayılar:** 0-9 rakamlarından oluşan, ondalık değerleri olmayan sayılardır. Örnek: `3` `5`

**Reel Sayılar:** tam ve ondalık kısımları olan sabitlerdir.

Örnek: `5.7` `4.2`




**Karakterler:** tek tırnak işareti arasında bulunur.

Örnek: `'A'` `'k'`

**Dizgiler:** Çift tırnak işareti " " ile tanımlanırlar ve birden fazla karakterin bir araya gelmesinden oluşurlar.

Örnek: "Merhaba" "2" "aa12aa"

# Veri Tipleri

| Veri Tipi | Açıklama | Bit | Aralık |
|---|---|---|---|
| `int` | Tamsayı (integer) | 16<br>32 | -32768 ile 32768<br>-4294967296 ile 4294967295 |
| `double` | Çift duyarlı reel sayı | 64 | Yaklaşık 12 basamak duyarlı |
| `char` | Karakter | 8 | 0 ile 255 |
| `short int` | Kısa tamsayı | 8 | -128 ile 127 |
| `long int` | Uzun tamsayı | 32 | -4294967296 ile 4294967295 |
| `unsigned int` | İşaretsiz tamsayı | 16 | 0 ile 65535 |
| `float` | Reel Sayı | 32 | Yaklaşık 6 basamak duyarlı |

# Değişkenler

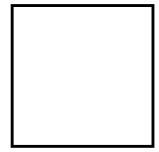
- Program içinde kullanılan temel nesnelerdir.
- Farklı tipteki verilerin (int, double, char) bellekte tutulması için kullanılır.
- Bu isimler ile bellek hücrelerine veri aktarabilir ve bu hücrelerde saklanan veriler program içinde kullanılabilir.

```
int toplam;
```

veri tipi

değişken ismi

toplam



Bellek görüntüsü

**Örnek:**

```
char cinsiyet;
double maas;
int x,y;
```