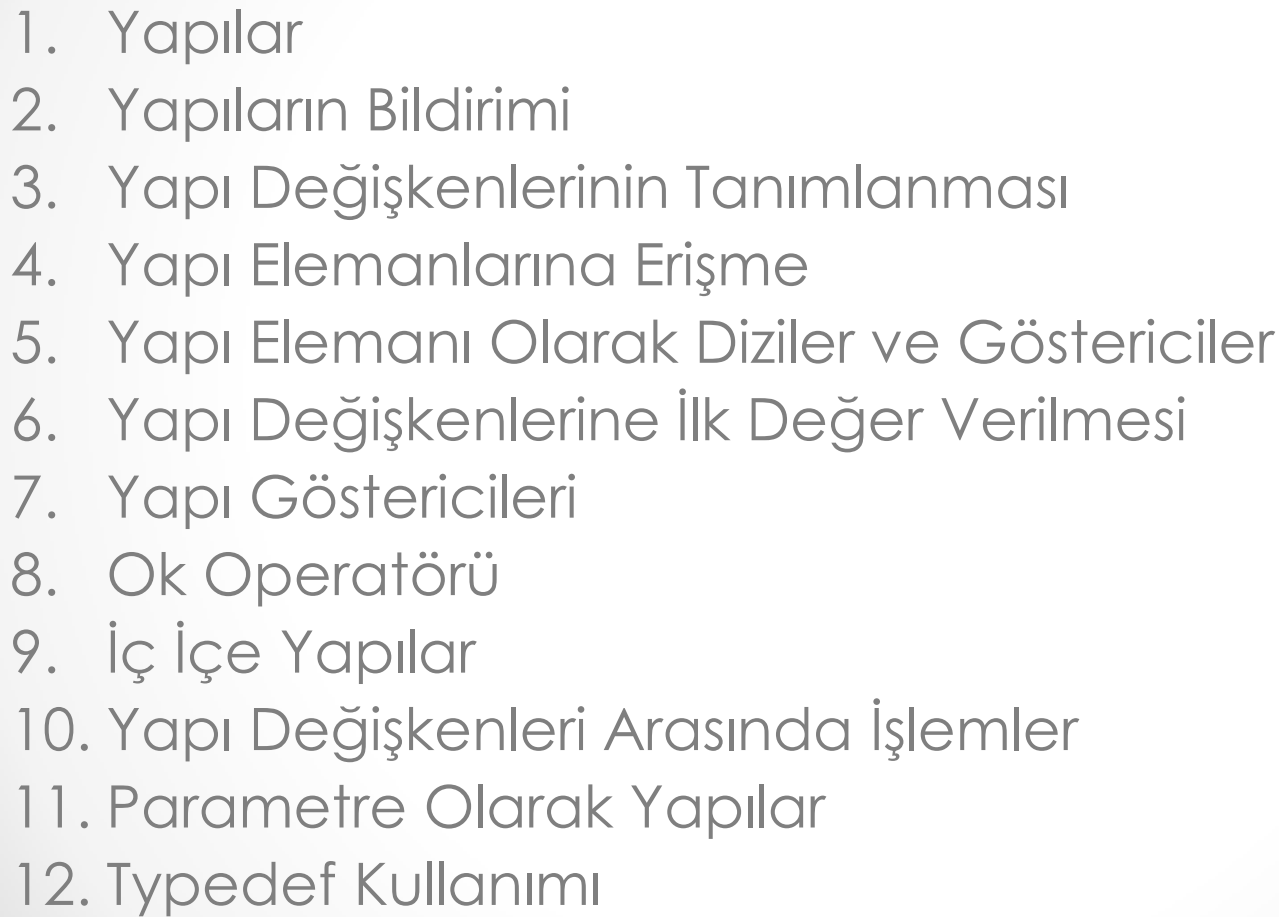




Dr. Zeynep Banu ÖZGER





# 1- YAPILAR

- Aralarında mantıksal bir ilişki bulunan, farklı türden bilgiler içeren veri yapılarıdır.
- Örneğin öğrenci kaydı;
  - Ad-soyad char, numara int, ort double
- Yapılar ve diziler benzerdirler.
- Yapılar
  - Diziler gibi belleğe bitişik konumlarda yerleşirler.
  - Diziler gibi başlangıç adresleri üzerinden erişilebilir ve fonksiyonlara gönderilebilir.





## 2- YAPILARIN BİLDİRİMİ

- Struct[yapı\_ismi]
- {
  - <tür> <yapı\_elemanı>;
  - <tür> <yapı\_elemanı>;
  - ...
- };
- Struct TARİH
- {
  - int gün;
  - int ay;
  - int yıl;
- }

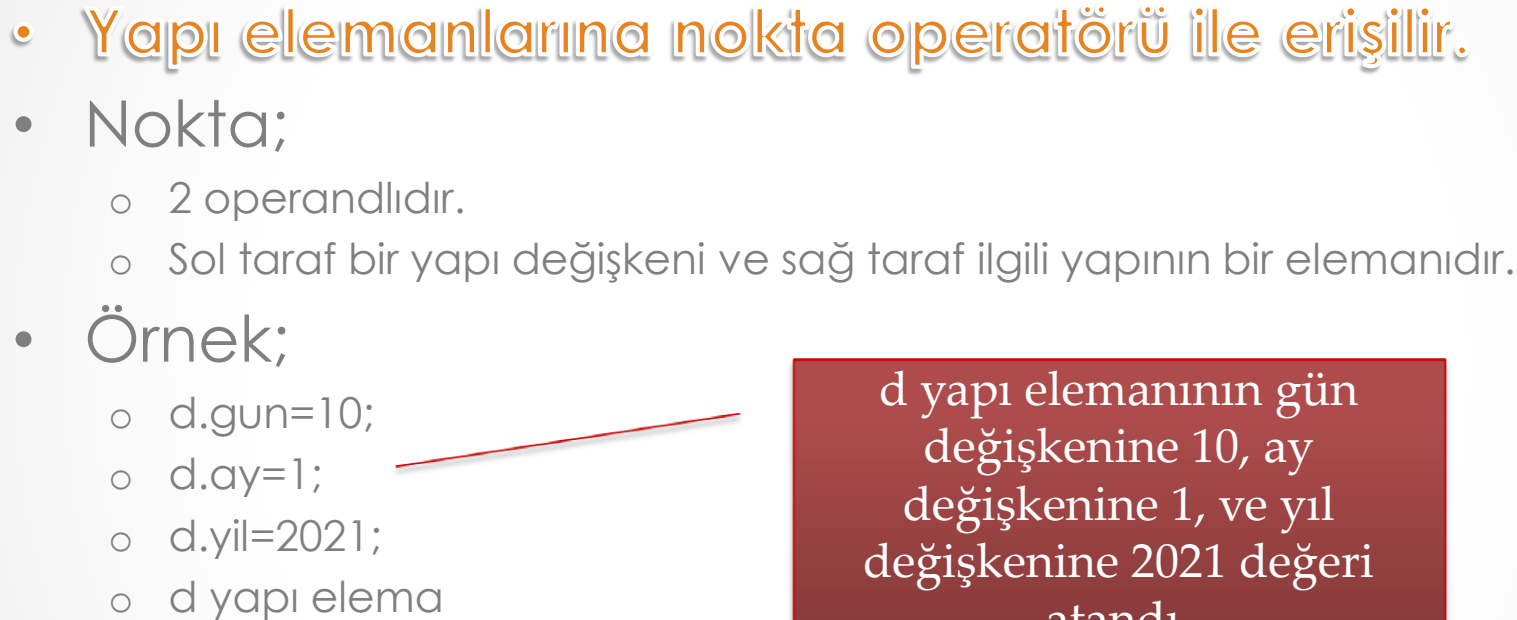
Tarih bilgisi 3 ayrı  
değişken yerine bir yapı  
ile tanımlanabilir.



- ```
o struct TARIH d;
```

## Yapı değişkeni tanımlama





d yapı elemanının gün  
değişkenine 10, ay  
değişkenine 1, ve yıl  
değişkenine 2021 değeri  
atandı.





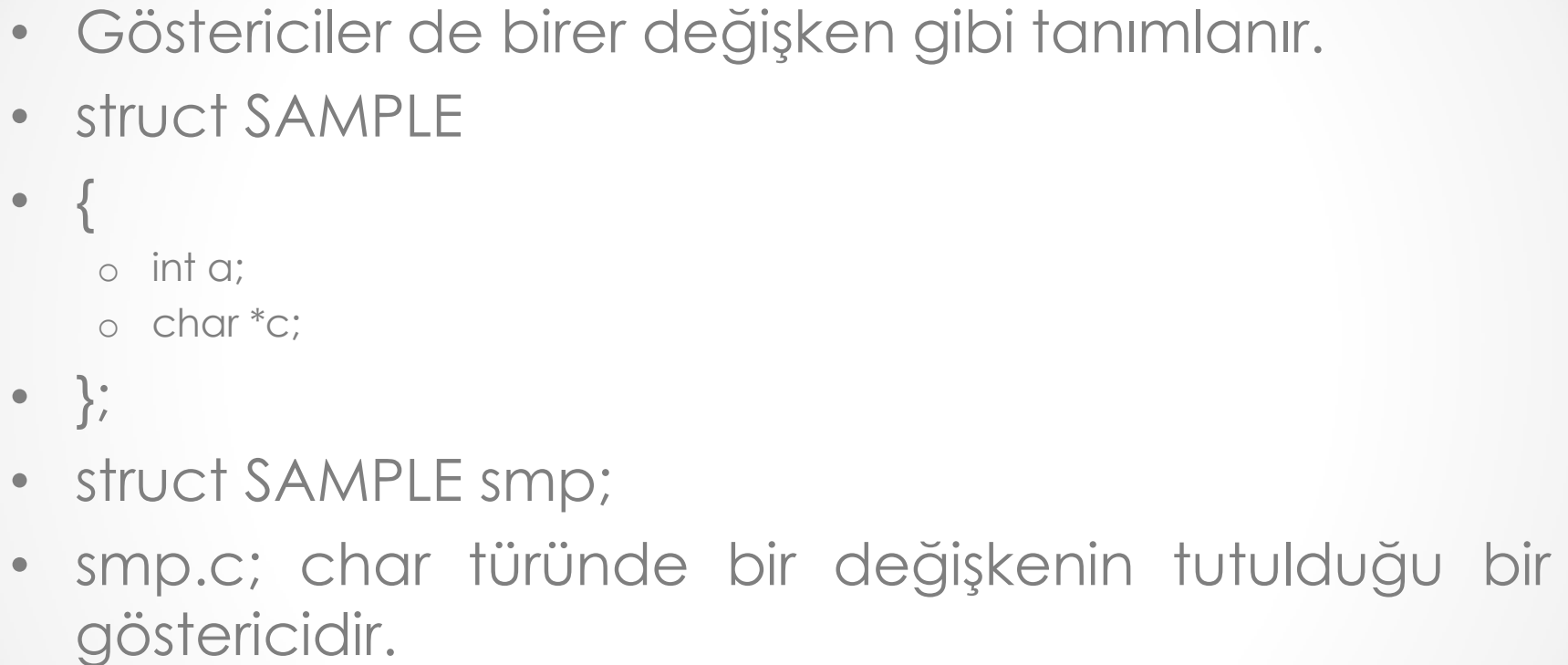
'TARİH' yapısı ile tanımlanmış günün tarihi ve doğum tarihine göre birinin yaşını hesaplayalım.



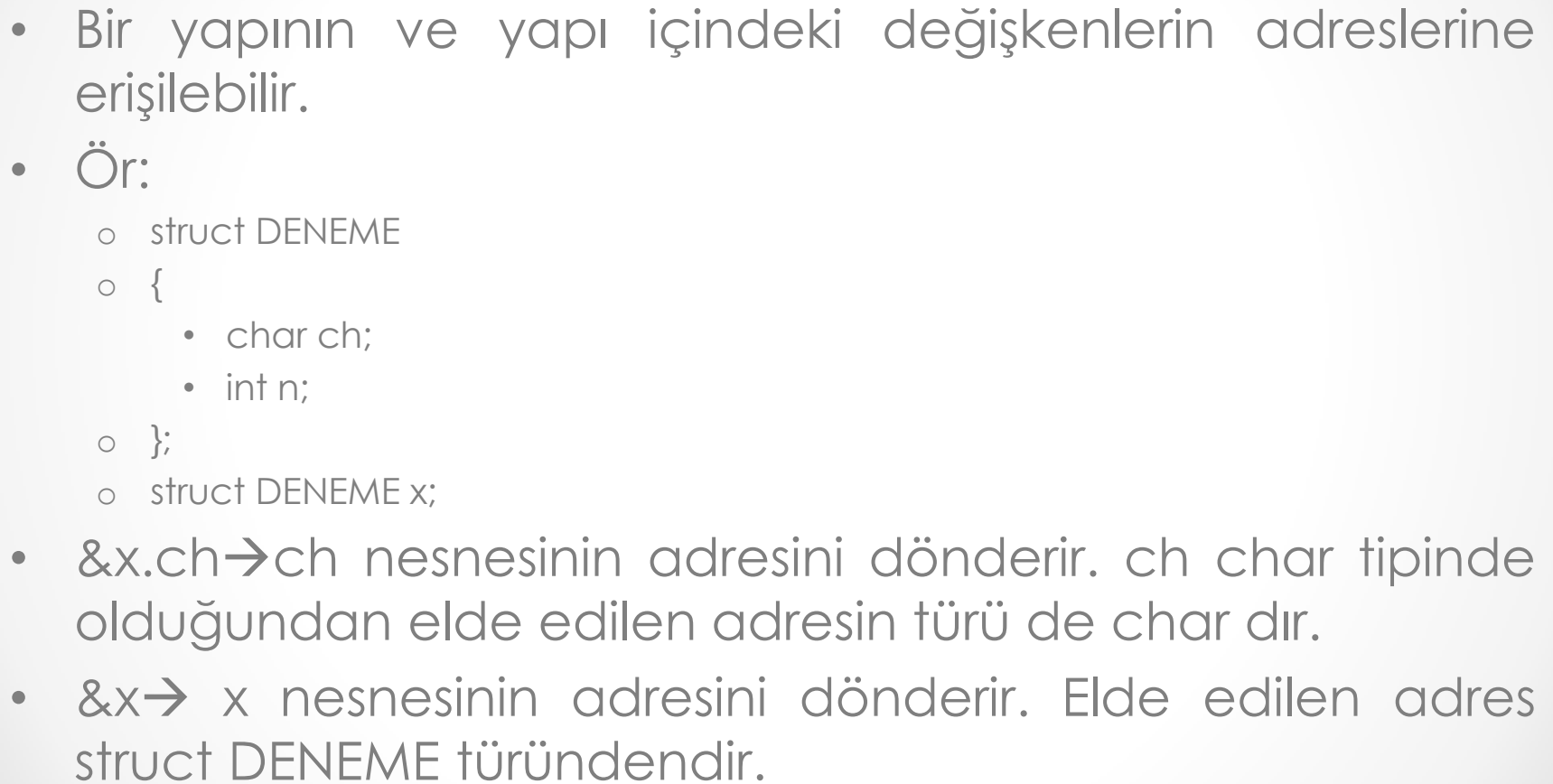
```
{
    char ad_soyad[30];
    char dogum_yeri[30];
    int yas;
};
```

```
struct INSAN x;
```

- Yapıdaki dizilerin isimleri, yapıdaki dizilerin bellekteki başlangıç adreslerini gösterir.
  - `x.ad_soyad` ve `x.dogum_yeri` ilgili dizilerin başlangıç adresleridir.
- Yapıdaki bir dizinin `n`. elemanına erişmek için;
  - `x.adi_soyadi[n]`









## 7- YAPI GÖSTERİCİLERİ

- Yapi göstericisi tanımlarken;
  - struct <yapi\_ismi> \*gosterici\_ismi;
  - Ör:
    - struct DENEME x; → x adında DENEME türünde yapı nesnesi tanımlandı
    - struct DENEME \*p; → p adında DENEME türünde yapı gösterici tanımlandı
    - p=&x; → yapının adresi göstericiye atandı
    - (\*p).n=5;

. operatörü \*  
operatöründen öncelikli  
olduğundan, Parantez ile  
önceliklendirilmelidir.



## 8- OK (->) OPERATÖRÜ

- Yapı göstericileri ile yapı nesnelere erişmek için kullanılır.
- 2 operandlıdır;
  - Sol taraftaki eleman yapı göstericisi sağ taraftaki yapı elemanıdır.
- (\*p).a ile p->a aynı anlamdadır.
- Ör:
  - void tarihGoruntule(struct TARİH \*x)
  - {
  - printf("Gun:%d\n",x->gun);
  - printf("Ay:%d\n",x->ay);
  - printf("Yıl:%d\n",x->yil);
  - }
- -> operatörü en öncelikli operatör grubundadır ve önceliği (), . ve [n] ile aynıdır.



- 







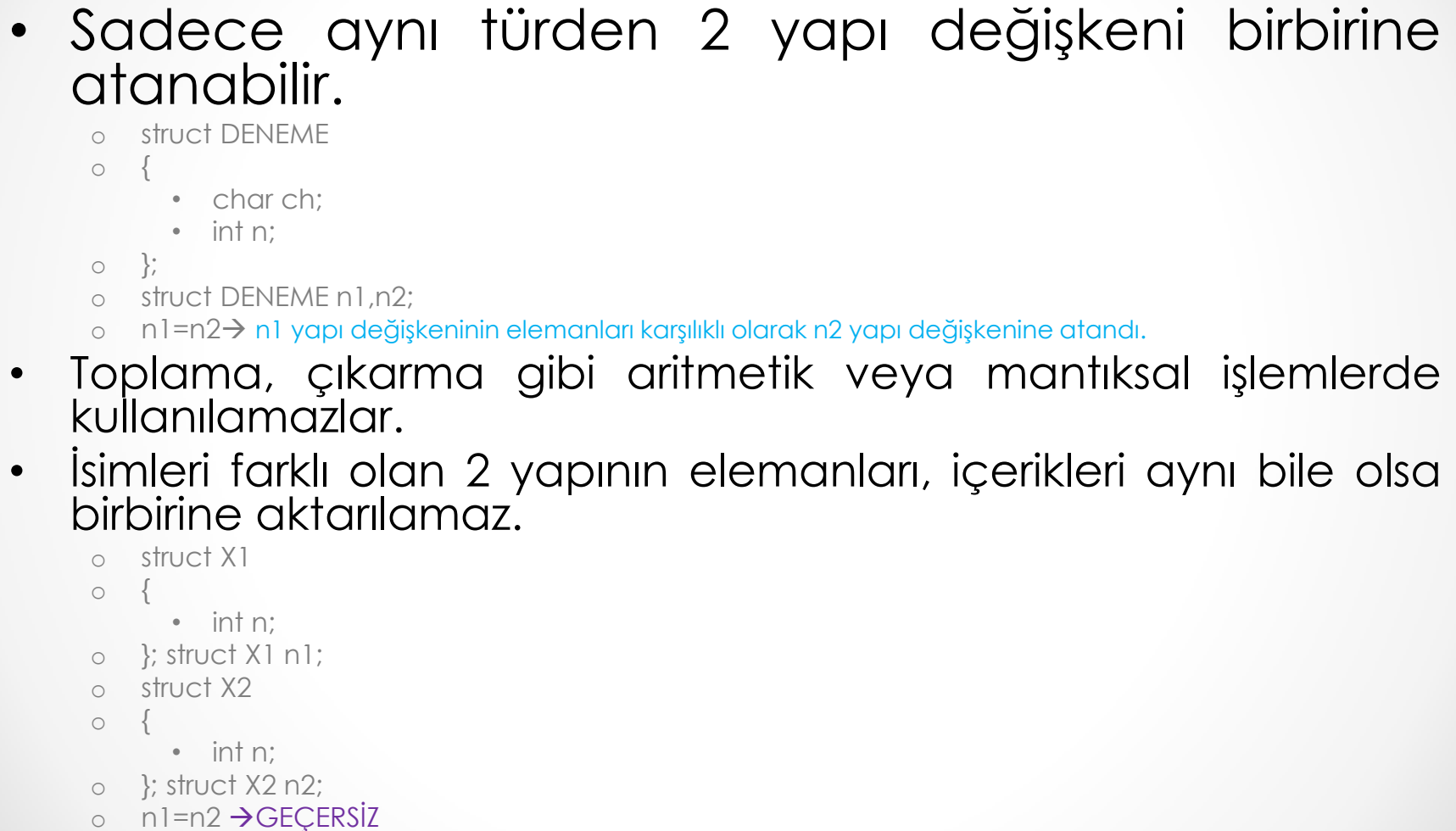
## 9- İÇ İÇE YAPILAR

- Bir yapı içinde başka bir yapı tanımlanmasıdır.
- 2 şekilde yapılabilir.
  - İçerdeki yapının bildirimi daha yukarda yapılır.
  - İçerdeki yapı dıştaki yapının içinde bildirimi yapılır.

```
struct TARIH
{
    int gun, ay, yil;
};

struct INSAN
{
    char name[30];
    struct TARIH dTarih;
};
```

```
struct INSAN
{
    char name[30];
    struct TARIH
    {
        int gun, ay, yil;
    }dTarih;
};
```





# 11- PARAMETRE OLARAK YAPILAR

- Yapılar, direk bir yapı veya adresleri ile parametre olarak fonksiyonlara geçirilebilir.

```
#include <stdio.h>
struct TARİH
{
    int gun, ay, yıl;
};

void tarihGoruntule(struct TARİH x)
{
    printf("Gun:%d\n", x.gun);
    printf("Ay:%d\n", x.ay);
    printf("Yıl:%d\n", x.yıl);
}

int main(void)
{
    struct TARİH n;
    n.gun=30.;
    n.ay=12;
    n.yıl=2021;
    tarihGoruntule(n);
}
```

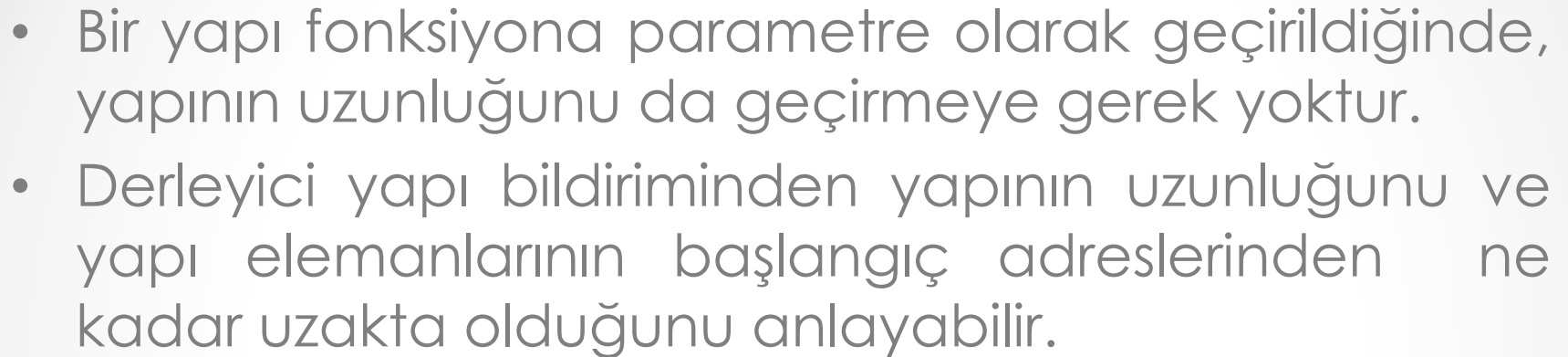


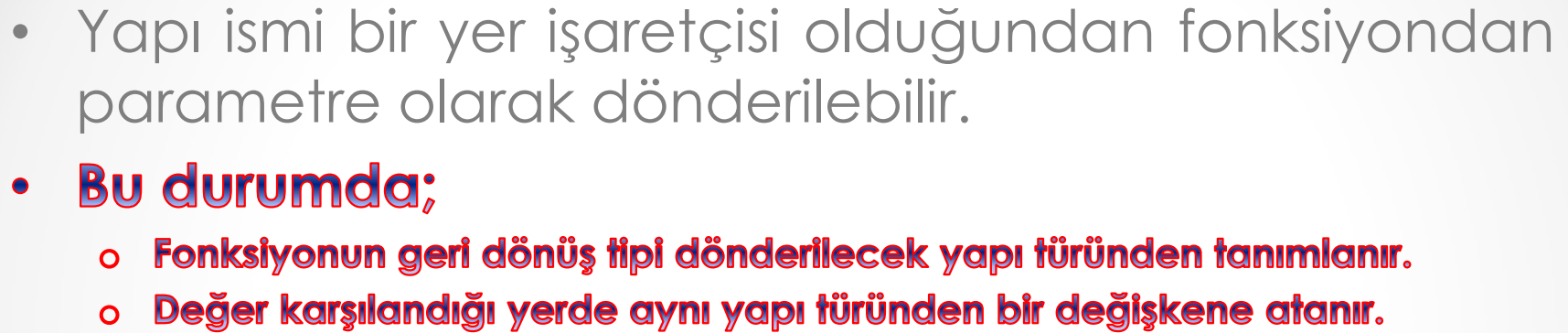
- Yapılar, direk bir yapı veya adresleri ile parametre olarak fonksiyonlara geçirilebilir.

```
#include <stdio.h>
struct TARİH
{
    int gun, ay, yıl;
};

void tarihGoruntule(struct TARİH *x)
{
    printf("Gun:%d\n", (*x).gun);
    printf("Ay:%d\n", (*x).ay);
    printf("Yıl:%d\n", (*x).yıl);
}

int main(void)
{
    struct TARİH n;
    n.gun=30.;
    n.ay=12;
    n.yıl=2021;
    tarihGoruntule(&n);
}
```







# 12- TYPEDDEF

- *Typedef; C'de var olan değişken türlerine yenisini eklemek için kullanılır.*
- *Ör:*
  - typedef int LENGTH;
  - C'de tanımlı int türünün aynısından 'LENGTH' ismiyle oluşturur.
- Struct tanımlamada da, tanım işlemlerini kolaylaştırmak için kullanılabilir.
- Aşağıdaki 2 kullanım eş değerdir.

```
struct TARIH
{
    int gun, ay, yil;
};

int main
{
    struct TARIH d1,d2;
}
```

```
typedef struct TARIH
{
    int gun, ay, yil;
}tarihler;

int main
{
    tarihler d1,d2;
}
```