

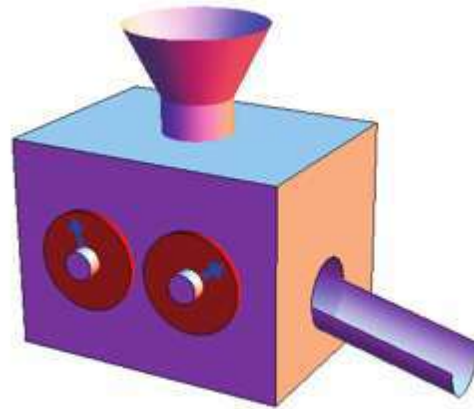
## 6. FONKSİYONLAR 1



- Bu bölümde;
- Fonksiyon kullanmanın gerekçelerini öğrenebilecek,
- Python programlama dilinde kullanılan standart fonksiyonları kullanabilecek,
- Fonksiyon türleri konusunda bilgi sahibi olabileceksiniz.

# 6.1. Neden Fonksiyonlar?

- Büyük ve kapsamlı bir program yazdığımızı ve program kapsamında pek çok kez aynı işlemi yapmamız gerektiğini düşünelim.
- Örneğin karekökü hesaplamak.



## 6.2. Fonksiyon Nedir?

### 6.3. Fonksiyonlar

- Bu kodu paketleyerek tekrar tekrar kullanmamızı sağlayan yaklaşımlardan biri “fonksiyonlar”dır.
- ***Bir fonksiyon, tekrar kullanılabilen kod parçacıdır.***
- **Kendimiz fonksiyon yazabileceğimiz** gibi önceden yazılmış ve kullanıma **hazır fonksiyonları** da kullanabiliriz.
- Python kapsamında da standart fonksiyonların bulunduğu bir **kütüphane** vardır.
- print, input, int, float, str, ve type.

## 6.4. Fonksiyon ve Modüller

- Bir Python modülü Python kodları içeren bir dosyadır.
- Dosyanın adı modülün adına işaret eder.
- Örneğin math.py isimli bir dosya standart matematik modülünde yer alan fonksiyonları içerir.

**from**

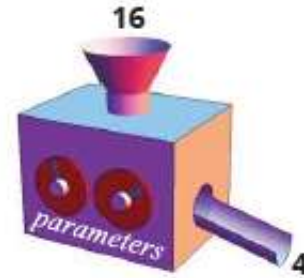
modül adı

**import**

fonksiyon adı

## 6.3.1. sqrt() Fonksiyonu

- Karekök hesaplar

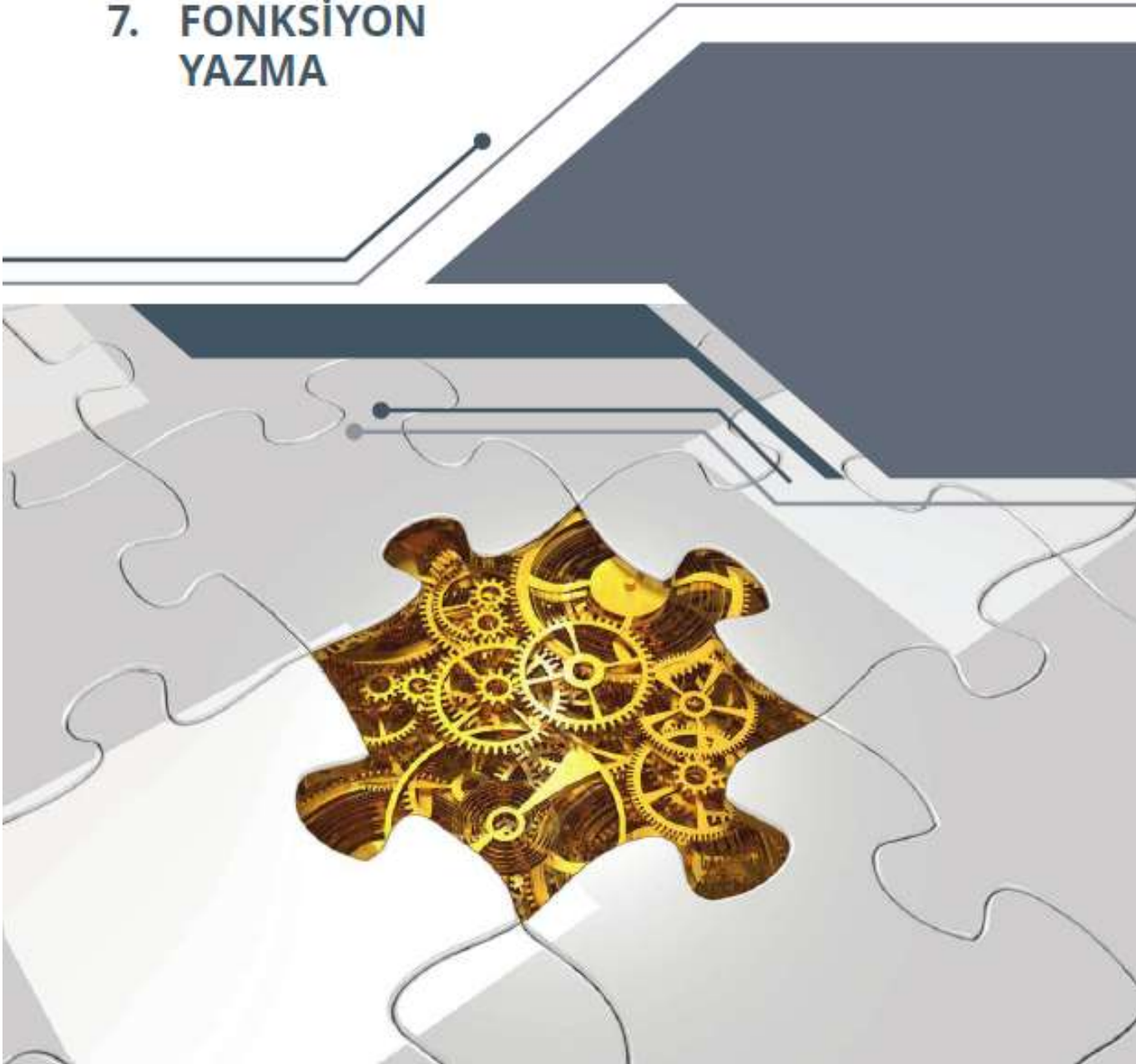


- **from math import sqrt**
- **# Kullanıcıdan değer alınıyor**
- **sayi = float(input("Sayı Giriniz: "))**
- **# Karekök hesaplanarak kök değişkenine aktarılıyor**
- **kok = sqrt(sayi)**
- **# Sonuçlar yazdırılıyor**
- **print(sayi, " sayısının karekökü" "=", kok)**

## *math Modülü*

Fonksiyon Adı	Açıklama	Örnek Kullanım ve Çıktı
<code>sqrt()</code>	İstenilen değerin karekök'ünün bulunmasını sağlar.	<code>math.sqrt(16)</code> → 4
<code>exp()</code>	e (Euler sabiti) sayısının istenilen kuvvetinin alınmasını sağlar.	<code>math.exp(2)</code> → 7.389056
<code>log()</code>	<code>log(x,y)</code> fonksiyonumuz iki parametre alır. İlk parametremiz olan x logaritması, alınacak sayıyı; ikinci parametre olan y taban sayısını temsil etmektedir.	<code>math.log(2,2)</code> → 1.0
<code>log10()</code>	<code>log(x,y)</code> fonksiyonundan tek farkı taban olarak 10 sayısının sabit olmasıdır.	<code>math.log10(10)</code> → 1.0
<code>cos()</code>	<code>cos(x)</code> , x derecesinin kosinüs değerini verir.	<code>math.cos(45)</code> → 0.5253
<code>pow()</code>	<code>pow(x,y)</code> fonksiyonu x sayısının y. kuvvetinin alınmasını sağlar.	<code>math.pow(2,2)</code> → 4
<code>degress()</code>	<code>degress(x)</code> fonksiyonu x açısını radyandan dereceye çevirmeye yarar.	<code>math.degress(45)</code> → 2578.310078088
<code>radians()</code>	<code>radians(x)</code> fonksiyonu x açısını dereceden radyana çevirmeye yarar.	<code>math.radians(45)</code> → 0.7853981633
<code>fabs()</code>	<code>fabs(x)</code> fonksiyonu x değerinin mutlak değerinin alınması işlemini gerçekleştirir.	<code>math.fabs(-5)</code> → 5.0

## 7. FONKSİYON YAZMA





## 7.1. Fonksiyon Kavramı

Program yazarken kod satırları uzayabilir ve yazılan program karmaşık bir hâl alabilir. Bu durumu ortadan kaldırmak için problemi alt problemler hâlinde ele almak ve fonksiyon yazmak gerekir. Böylesine bir çözüm yolu birçok yarar sağlar:

- Programın yönetimi kolaylaşır.
- Daha doğru çözüm üretilebilir.
- Daha kolay hata ayıklama yapılabilir.
- Kod satırlarını değiştirmek/genişletmek kolaylaşır.

Python programlama dilinde bir fonksiyon için iki durum söz konusudur:

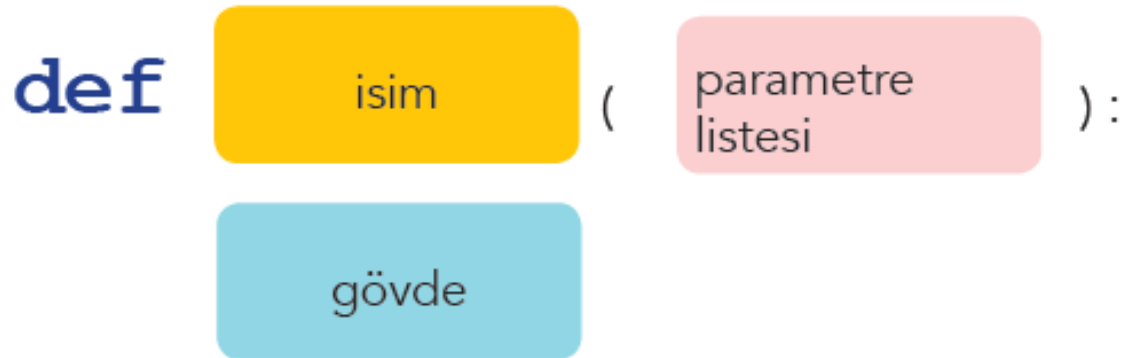
- Fonksiyon tanımlama: Fonksiyonun nasıl davranacağını tanımlayan kod satırları.
- Fonksiyon çağırma: Program içinde fonksiyonun çağırılması ile kod satırlarının çalışması.

Her fonksiyonun bir kez tanımlanması ancak farklı şekillerde çağırılması söz konusudur.

## 7.1.1. Fonksiyon Tanımlama

Fonksiyon tanımlamak için dikkat edilmesi gereken dört durum vardır:

- def: Bu ayrılmış sözcük ile fonksiyon tanımlama başlar.
- İsim: Fonksiyon için bir isim verilmelidir. Aynı değişken tanımlamada olduğu gibi.
- Parametre: Fonksiyon içinde kullanılan değerleri ifade eder.
- Gövde: Fonksiyon için gerekli olan kod blokundan oluşur.



## 7.1.2. Fonksiyon Yazma

Aşağıdaki örnekte def kelimesi fonksiyon tanımlama için kullanılmıştır. Fonksiyon ismi double olarak belirlenmiştir. Kullanıcıdan n ile bir değer istenmiştir. Fonksiyona ait kod bloku bir satırdan oluşmakta olup bu, girinti ile ötelenmiştir.

```
def double(n):  
    return 2 * n  
  
x = double(3)  
print(x)
```

## 7.1.3. Fonksiyon Çağırma

- Aşağıdaki örnekte fonksiyon, 5 değeri ile a değişkeni içerisine çağırılmıştır. return komutu ile gelen değer, print() komutu ile yazdırılmıştır.

```
# karenin alanını hesaplayan program
def Alan(a)
    return a*a
hesapla=Alan(5)
print(hesapla)
```

## 7.2. Fonksiyon Kullanımı Örnekleri

### Örnek

```
def say():
    for i in range(1, 11):
        print(i, end=" ")
    print()
print("10"a kadar sayılıyor. . .")
say()
print("Tekrar 10"a kadar sayılıyor. . .")
say()
```

### Ekran Çıktısı

```
10"a kadar sayılıyor. . .
1 2 3 4 5 6 7 8 9 10
Tekrar 10"a kadar sayılıyor. . .
1 2 3 4 5 6 7 8 9 10
```