



[www.youtube.com/BMdersleri](http://www.youtube.com/BMdersleri)

# MAKÜ

BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Flutter ile Mobil Programlamaya Giriş



## 6.HAFTA FONKSİYONLAR

1



Hazırlayan	: Zeynep İrem KESLER 1911404048
Tarih	: 05/04/2022
Sürüm	: v1
Ders Yürütücüsü	: Doç. Dr. İsmail KIRBAŞ

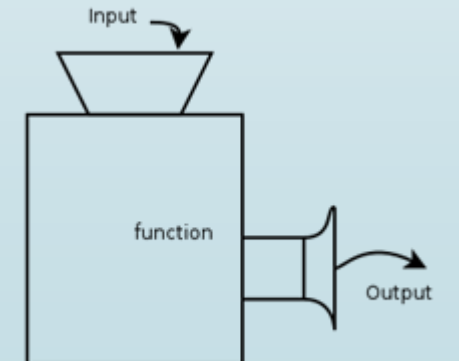
# İÇİNDEKİLER

- Fonksiyon Nedir?
- Fonksiyon Tanımlama
- Geriye Değer Döndürmeyen Fonksiyonlar
- Parametrelili Fonksiyonlar
- Opsiyonel Parametrelili Fonksiyonlar
- Geriye Değer Döndüren Fonksiyonlar
- İsimli Parametre Alan Fonksiyonlar
- Arrow Fonksiyonlar
- Anonim Fonksiyonlar
- Asenkron Fonksiyonlar
- Simple Function in Flutter
- Yardımcı Kaynaklar



# Fonksiyon Nedir?

- Fonksiyonlar diğer programlama dillerinde de olduğu gibi, dart dilinde de kullanılan temel yapılardan biridir.
- Fonksiyonlar temelde belirli bir amaç için oluşturulmuş, tekrar tekrar kullanılabilen ve mantıksal bütünlük gösteren kod bloklarıdır.
- Fonksiyonlar sayesinde yazılan kodun okunabilirliği kolaylaşır, üzerinde değişiklik yapmak basittir, tekrar tekrar çağırılabilirdiği için kod kalabalığını azaltır.



# Fonksiyon Tanımlama

```
fonksiyonAdi() {  
    Görevler  
}
```

**NOT:** Yazılan fonksiyonlar çağırılmadığı sürece program içinde etkisizdir.

Örnek:

```
pencere () {  
    print("Pencere açıldı.") ;  
}  
  
void main() {  
    pencere();  
}
```

Çıktı:

Pencere açıldı.

# Geriye Değer Döndürmeyen Fonksiyonlar

Geriye değer döndürmeyen fonksiyonlar işlemleri kendi içinde yapar ve sonucu yine kendi içinde oluşturur. Geriye herhangi bir değer döndürmez. Bu tip fonksiyonları oluştururken fonksiyon isminin başına '**void**' ifadesi yazılır. Bu ifade fonksiyonun değer döndürmediğini belirtir. Eğer dikkat edildiyse program kodlarının çalıştığı '**main**' geriye değer döndürmeyen void türünde bir fonksiyondur.

```
void carpma() {  
    int sonuc;  
    sonuc = 5*10;  
    print(sonuc) ;  
}  
void main() {  
    carpma() ;  
}
```

Çıktı:

50

# Parametrelili Fonksiyonlar

Parametrelili fonksiyonlar, fonksiyon yazıldıktan sonra yanında açılan parantezlerle değeri alan fonksiyonlardır. Bu fonksiyona parametreleri, programda çağırıldığı yerde verilir.

```
void main() {  
    toplama(10,12);  
}  
  
toplama(int sayi1, int sayi2) {  
    print(sayi1+sayi2);  
}
```

Çıktı:

22

# Opsiyonel Parametrelili Fonksiyonlar

Parametrelili fonksiyonların alacağı parametre ya da parametreler isteğe bağlı olarak yollanır. Eğer parametreye değer yollanmazsa varsayılan değeri ne ise o değeri kullanır. İsteğe bağlı olarak yollanacak parametre ya da parametreler **[]** arasına yazılır. Eğer isteğe bağlı parametreye değer gönderilmezse ya da parametreye varsayılan değeri ataması yapılmazsa parametre **'null'** değeri alır. Zorunlu değeri alacak parametre her zaman önce yazılır.

```
void main() {  
    bolme(50);  
}  
  
void bolme(int sayi1 , [int sayi2=20]){  
    double sonuc;  
    sonuc=sayi1/sayi2;  
    print(sonuc);  
}
```

Çıktı:

2.5

# Geriye Değer Döndüren Fonksiyonlar

Görevini yerine getirdikten sonra geriye bir sonuç teslim eden fonksiyonlardır. 'return' komutu kullanılarak geriye sonuç döndürür. Fonksiyonun döndürdüğü veri tipi ne ise fonksiyonun başına o veri tipinin yazılması tavsiye edilir. Eğer yazılmazsa otomatik olarak 'dynamic' türünde olacaktır.

```
void main() {  
    print(ortalama());  
}  
  
double ortalama() {  
    double sonuc;  
    sonuc = (50+60+70) / 3;  
    return sonuc;  
}
```

Çıktı:

60



# Geriye Değer Döndüren Fonksiyonlar

```
void main() {  
    print(birlestir("Ali" , "Yıldız"));  
}  
  
String birlestir(String adi, String soyadi)  
{  
    String adiSoyadi;  
    adiSoyadi = adi+ " " + soyadi;  
    return adiSoyadi;  
}
```

Çıktı:

Ali Yıldız

# İsimli Parametre Alan Fonksiyonlar

Fonksiyonların parametrelerine değer yollanırken isimlerini kullanarak yollama yöntemidir. Bunun için fonksiyonun parametreleri {} içerisinde yazılır.

**Fonksiyon\_Adi({parametre1,parametre2...}) {Komutlar ; }**

```
void main() {  
    print(kalanBul(sayi1:10 , sayi2:3));  
}  
  
int kalanBul ({required int sayi1 , required int sayi2}) {  
    int kalan;  
    kalan = sayi1 % sayi2 ;  
    return kalan;  
}
```

Çıktı:

1

# Arrow Fonksiyonlar

- Bu yöntemle fonksiyon tek satırda yazılabilmektedir.

**Fonksiyon\_Adi()=> Komutlar ;**

- '='>' işareti return komutunun görevini yapar.

```
void main() {  
    print(topla(8,15));  
}
```

```
int topla(int sayi1, int sayi2) => sayi1 + sayi2 ;
```

Çıktı:

23

# Anonim Fonksiyonlar

İsmi olmayan fonksiyonlardır. Anonim fonksiyonları çağırmak için değişkenler kullanılabilir.

**(){Komutlar;} veya ()=> Komutlar ;** şeklinde oluşturulur.

```
void main() {  
    fonksiyon();  
}
```

```
Function fonksiyon = () => print("Ben bir anonim  
fonksiyonum");
```

Çıktı:

Ben bir anonim fonksiyonum

# Asenkron Fonksiyonlar

Asenkron fonksiyonlar, gerçekleşmesi uzun zaman alan işlemlerin gerçekleşmesini beklerken, farklı görevdeki işlemler varsa bu işlemlerin aynı anda gerçekleştirilebilmesini sağlayan fonksiyonlardır.

Asenkron fonksiyonlarda bazen bir işlem sonucu gerçekleşmeden başka işleme geçilmez. Bu durumda fonksiyon içinde beklenmesi gereken komut satırının başına **'await'** komutu eklenir. Bu komutu kullanabilmek için de fonksiyon isminin başına **'future'**, fonksiyon isminden sonra **'async'** komutu eklenir. Eğer fonksiyon geri değer döndüren bir fonksiyonsa, **'future'** kelimesinden sonra **'< >'** yapısının içine geri değer dönüş türü yazılır. Böylelikle fonksiyon bitirilmesi gereken işlemin bitmesini bekler ve bir sonraki komut satırına öyle geçer.

```
Future<void> main() async{
    print(await kahveSoyle());
}

Future<String> kahveSoyle() async{
    var kahveDurumu = await kahveHazirla();
    return "KAHVE DURUMU: $kahveDurumu";
}

Future<String> kahveHazirla() {
    return Future.delayed(Duration(seconds: 3), () => "KAHVENİZ HAZIRLANIYOR");
}
```

Çıktı:

KAHVE DURUMU: KAHVENİZ HAZIRLANIYOR

# Simple Function in Flutter

```
void printText (String text) {  
    print(text) ;  
}
```

- İşlev aracılığıyla bir şey döndürüyorsanız, işlev adından önce veri türünü belirtmeniz gerekir.

# Simple Function in Flutter

Dart Defteri    Yeni Ped    Sıfırla    Biçim    SDK'yı y

```
1 import 'package:flutter/material.dart';
2
3 void main() => runApp(MyApp());
4
5 class MyApp extends StatelessWidget {
6
7   String printText (String text) {
8     print(text);
9     return 'Hello $text';
10  }
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Welcome to Flutter',
15      home: Scaffold(
16        appBar: AppBar(
17          title: Text('Welcome to Flutter'),
18        ),
19        body: Center(
20          child: Text(printText('World')),
21        ),
22      ),
23    );
24  }
25 }
26
```

Koşmak

Welcome to Flutter

DEBUG

Hello World

# Yardımcı Kaynaklar

- Adım Adım Flutter İle Mobil Uygulamalar ( Rakıcı Oğuz , 2021)







[www.youtube.com/BMdersleri](http://www.youtube.com/BMdersleri)

**MAKÜ**  
BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Flutter ile Mobil Programlamaya Giriş



# İlginiz için teşekkürler...

17



Hazırlayan  
E-posta

: **Zeynep İrem KESLER 1911404048**  
: zeynepiremkesler@gmail.com

Tarih

: 05/04/2022

Ders Yürütücüsü

: Doç. Dr. İsmail KIRBAŞ

E-posta

: ismkir@gmail.com