



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

**MAKÜ**

BURDUR MEHMET AKİF ERSOY ÜNİVERSİTESİ

Flutter ile Mobil Programlamaya Giriş



# 8.HAFTA

## Projeye İlk Bakış

## Widget Kavramı

## Ekran Oluşturma ve Scaffold

1



Hazırlayan	: Zeynep İrem KESLER 1911404048
Tarih	: 19/04/2022
Sürüm	: v1
Ders Yürütücüsü	: Doç. Dr. İsmail KIRBAŞ

# İÇİNDEKİLER

- Projeye İlk Bakış
- Flutter'ın Genel Özellikleri
- Widget Kavramı
- Material App
- Scaffold
- Ekran Oluşturma
- Yardımcı Kaynaklar



# Projeye İlk Bakış

## Flutter'ın Genel Özellikleri:

- **Flutter**, esnek bir SDK'dır ve Windows, Linux ya da Mac ortamlarında rahatlıkla çalıştırılabilir.
- **Flutter**, Mobile-first 2D render motoru, react-style framework'ü, Android ve iOS platformunda kullanılan widget desteğine sahip zengin bir SDK'dır.
- **Flutter**, yapısında Material Design ve Cupertino (iOS tarzı) widgetlar, temalar yardımıyla her iki platform için de efektif uygulamalar geliştirmeye olanak sağlar.
- **Flutter**, birçok geliştirme ortamında bulunmayan **Hot Reload** adında bir gerçek zamanlı düzenleme özelliğine sahiptir. Hot Reload ile geliştiriciler uygulamalarını hazırlar veya düzenlerken, gerçek zamanlı olarak yaptıkları değişikliklerin uygulama üzerindeki yansımalarını da anında görebilir.
- Flutter ile ilgili en güncel dökümanlara ulaşmak için: <https://flutter.dev/>
- Programlarda kullanılacak kütüphanelere ulaşmak için: <https://pub.dev/>

# Widget Kavramı

1. Flutter'ın sloganı, " Every thing is a Widget " yani " Her şey bir Widget " tır. Widgetları puzzle parçalarına benzetebiliriz. Puzzle'ın her bir parçası nasıl önemliyse her bir widget da o kadar önemlidir. Flutter yapısı içinde bulunan hemen hemen her şey **widget** dediğimiz yapıya sahiptir. Bu yapıyı ağaca benzetebiliriz. Ağacın gövdesi, ona bağlı dalları ve yaprakları gibidir. İşte tüm bunlar birleştğinde **widget ağaçları** oluşur ve bu da görsel yapıya dönüşür. Bu görsel yapı oluşturulurken; text widgetları, form widgetları ve sayamadığımız bir çok widget kullanılabilir.



**NOT:** Widgetlar ayrı ayrı kullanılabildiği gibi iç içe de kullanılabilir. Tıpkı Legolardan şekil türetmek gibi düşünülebilir.

# Widget Kavramı

**runApp:** Uygulamaya başlama standardıdır. Ağaç gibi düşünülebilir. Altındakiler ise ağacın dalları olarak düşünülebilir.

**MaterialApp:** runApp'ın altında çalıştırılan ilk standart widget'tır.

**Home:** Bu parametre bir widget alır. Bu widget kullanıcıya ilk gösterilecek olan arayüz nesnesidir. Bir sayfa ya da sıradan bir widget olabilir.

**Scaffold:** Bu widget oluşturulacak sayfaların temel iskeletidir. Sayfanın gövdesinde neler bulunacaksa burada oluşturulur.

**AppBar:** Uygulamaların üst kısmında yer alan ve genellikle uygulama veya bulunulan sayfanın isminin yer aldığı yapıdır.

**Body:** Gövdenin tamamını oluşturur.

**Material:** Materyalleri oluşturur. Renk, yazı tipi vs.

**NOT:** Yukarıdaki widget'lar temel widget'lardır. Bir widget'ın alacağı parametrelerin tümü görülmek isteniyorsa, widget ismi yazılıp parametre alacağı parantezleri açıp kapatıldıktan sonra parantezler arasına tıklayıp klavyeden " **ctrl + space** " tuşlarına aynı anda basılır. Kullanılacak widget'la ilgili ayrıntılı bilgi alınmak isteniyorsa, widget üzerine tıklanıp klavyeden " **ctrl + q** " tuşlarına basılır.

# Widget Kavramı

## Material App:

Material App uygulamanın ana ve en kapsayıcı olan alanıdır. Material tarzdaki geliştirmelerimiz için bize tüm widgetların dönüşünü veren en temel widgetlardan biridir.

```
import 'package:flutter/material.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        // This is the theme of your application.
        //
        // Try running your application with "flutter run". You'll see the
        // application has a blue toolbar. Then, without quitting the app, try
        // changing the primarySwatch below to Colors.green and then invoke
        // "hot reload" (press "r" in the console where you ran "flutter run",
        // or simply save your changes to "hot reload" in a Flutter IDE).
        // Notice that the counter didn't reset back to zero; the application
        // is not restarted.
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: MyHomePage(title: 'Merhaba Flutter'),
    ); // MaterialApp
  }
}
```

# Widget Kavramı

## Title:

Material App metodunun içerisine girip gerekli parantezleri açtığımızda karşımıza öncelikle **Title** çıkar. Tam olarak anlamını karşılayan Title widgetı uygulamanın adını içinde barındıran ve string bir değer alan yapıdır.

## Theme:

Uygulamamızın temasını burada seçeriz. Dark, Light ya da diğer seçeneklerin hepsini burada görebilirsiniz.

## Home:

Kullanıcı olan kişinin sayfada gördüğü her özelliğin çizdirilmesini tetikleyen widgettır. Kısacası Material App'in can damarıdır diyebiliriz. Ve bu home yapısının içerisinde tüm tetikleri gerçekleştiren tanıdık bir widget daha bulunur. **Scaffold**.

## Color:

Oluşturduğumuz sistemin arayüzünde uygulama için kullanacağımız birincil rengi belirlediğimiz kısımdır. Renklerin rengi diyebiliriz sanırım.(Buradan farklı renk seçenekleri kullanabilirsiniz. <https://www.materialpalette.com/>)

## Initialroute:

Bu widgetın içerisinde uygulamaya girdiğimizde karşımıza gelmesini istediğimiz ilk sayfanın **route** bilgisini veririz. Bizim için anasayfanın adresini taşır.

## Routes:

Tasarlanan diğer sayfaların yollarını tanımladığımız kısımdır. Her sayfa doğru ve hızlı bir erişim için adlandırılır ve bu özelliğe sahip sayfalara Named sayfalar denir.

## Locale:

Uygulamanın başlangıç dili ve ülke koduna ait bilgileri barındırır. Boş bırakılırsa da sistemin localini kullanır.

# Widget Kavramı

```
MaterialApp({ title, color, theme, initialRoute, routes, locale, localizationsDelegates, supportedLocale
//uygulama başlık renk tema.....
    title: 'Uygulama Adi',
    color: Colors.blue, //Sistemin arayüzünde uygulama için kull
    theme: ThemeData( //Uygulama için varsayılan renkler, yazı t
        primaryColorLight: Colors.blue, //Uygulamanın varsayılan
        brightness: Brightness.dark, //Uygulama varsayılan par
        backgroundColor: Colors.green, //Uygulamanın varsayılan
    ),
//Rout.....
    initialRoute: "/", // varsayılan ilk açılacak ana sayfa ,
    routes: {
        '/sayfabir': (context) => new SayafaBir(),
        '/sayfaiki' : (context) => new SayafaIki(),
        '/sayfauc' : (context) => new SayafaUc(),
        '/sayfadort' : (context) => new SayafaDort()
    },
//localizasyon.....
    locale: Localizations.localeOf(context); //Başlangıç yerel ayarı.Eğer 'yerel'
    localizationsDelegates: [
        AppLocalizationsDelegate(), // bizim delegemiz
        GlobalMaterialLocalizations.delegate,
        GlobalWidgetsLocalizations.delegate
    ],
    supportedLocales: const [Locale('en', 'US'), Locale('es', 'ES'), Locale('tr', 'TR')]
    onGenerateTitle: (BuildContext context) => AppLocalizations.of(context).title, // biz
//içerik.....
    home: Scaffold({ appBar: body: })),
// RouteFactory onGenerateRoute,
```

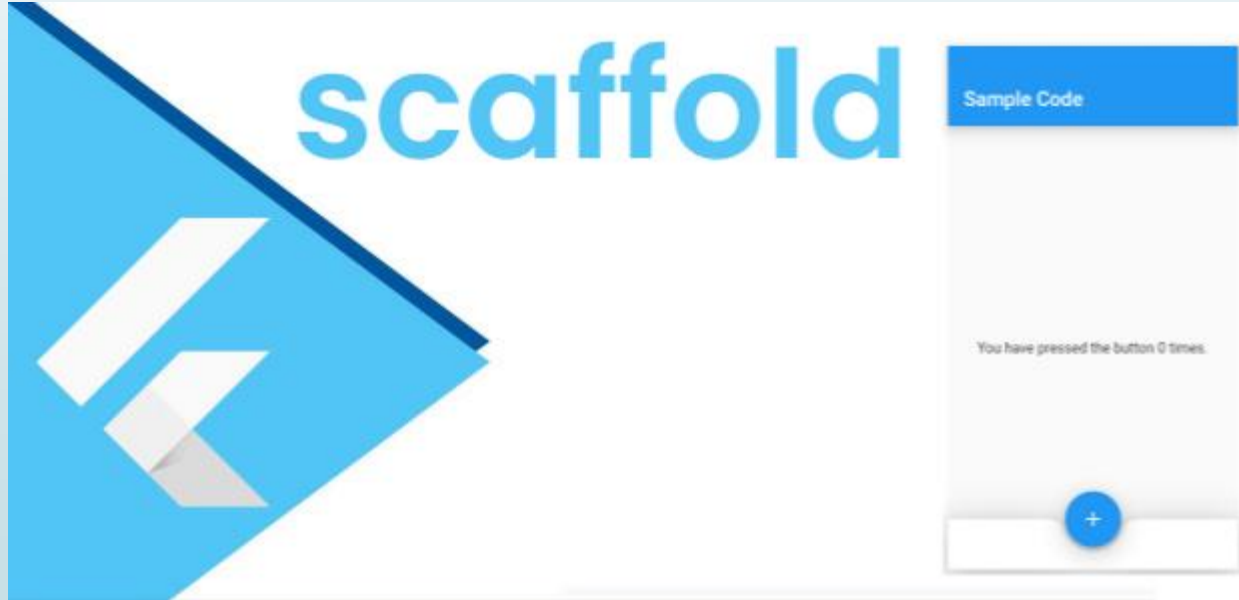


# Scaffold

Materyal dizayna uygun olarak tasarladığımız uygulamalarımızda yeni bir ekran oluştururken Scaffold widgetını kullanırız.

Ekrandaki öğelerin yerleşimini ayarlamak için en dışta scaffold kullanıyor olmamızın sebebi scaffold widgetının bize bazı avantajlar sağlıyor olmasıdır.

Scaffold widgetı uygulamamızın genel iskeletini temsil ediyor diyebiliriz.



# Scaffold

Scaffold, Flutter'ın içinde bulunan bir sınıftır. Bu sınıf AppBar, Bottomnavigationbar, Drawer gibi bir çok özelliği içinde barındırır.

## **AppBar:**

AppBar içinde bulundurduğu actions, leading ve bottom parametleriyle kişiselleştirebilir, esnek bir yapıya sahiptir.

## **Body:**

Sayfanın gövde kısmıyla alakalı yapmak istediklerimizi burada gerçekleştiririz. Yukarıdaki basit örnekte sadece gövde kısmının ortası alınarak 'Hello World' yazılmıştır.

## **Background Color:**

appBar ve body yapısının arka kısımdaki rengini belirler.

## **Drawer:**

Açılabilir bir menü yapısıdır. Mobil uygulamalarda sıkça karşılaşılan bir yapı olmasının yanında içerisinde liste, resim gibi çok çeşitli componentleri bulundurduğu için aslında elle baştan yapılması çok uzun sürer. Ama Flutter bunu bize hazır olarak sunar.

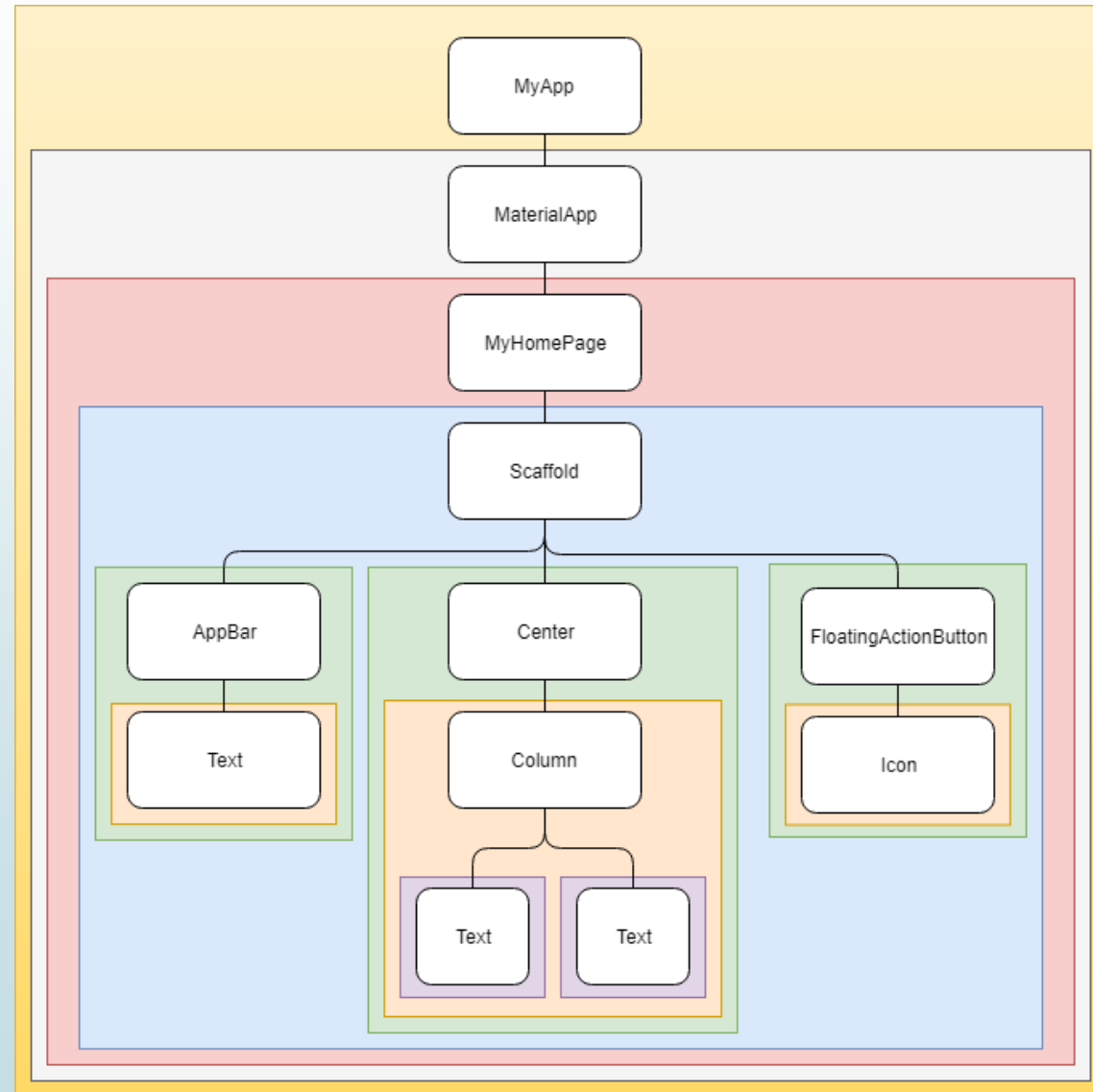
## **Bottom Navigation Bar:**

Ekranın alt kısmında bir menü oluşturmaya yardımcı olur. Bize kolay geçişler sağlar.

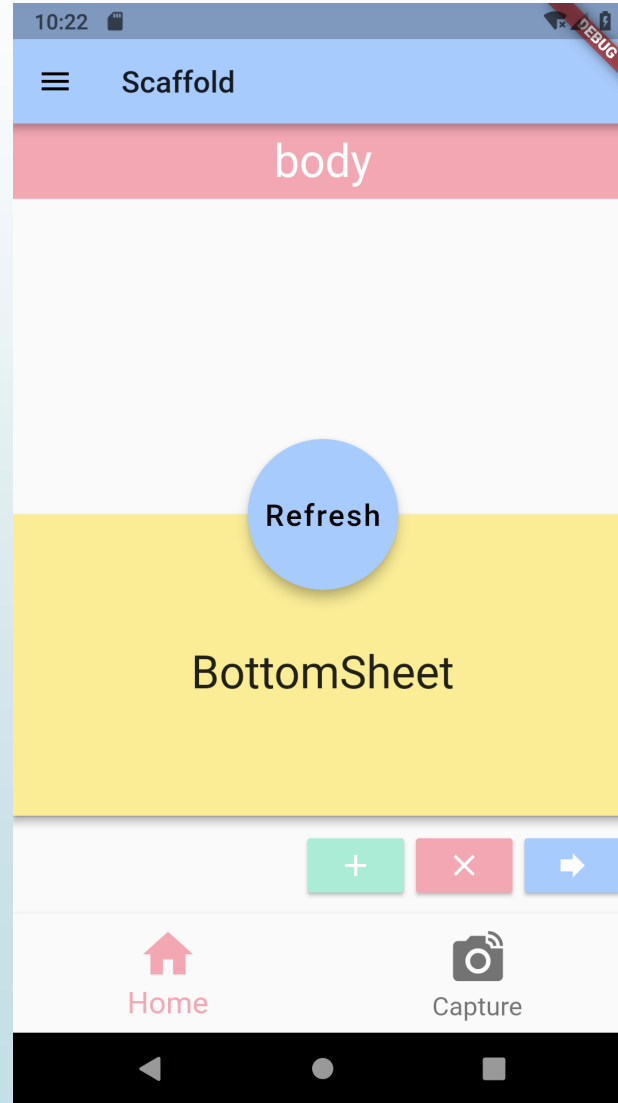
## **Floating Action Button:**

Ekranı yer değiştirebilen bir buton oluşturmamıza yarar. Location özelliği ile nereye gitmesi gerektiğini söyleyebiliriz.

# Ekran Oluşturma



# Ekran Oluřturma



# Yardımcı Kaynaklar

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





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



Flutter ile Mobil Programlamaya Giriş



# İlginiz için teşekkürler...

14



Hazırlayan  
E-posta

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

Tarih

: 19/04/2022

Ders Yürütücüsü

: Doç. Dr. İsmail KIRBAŞ

E-posta

: ismkir@gmail.com