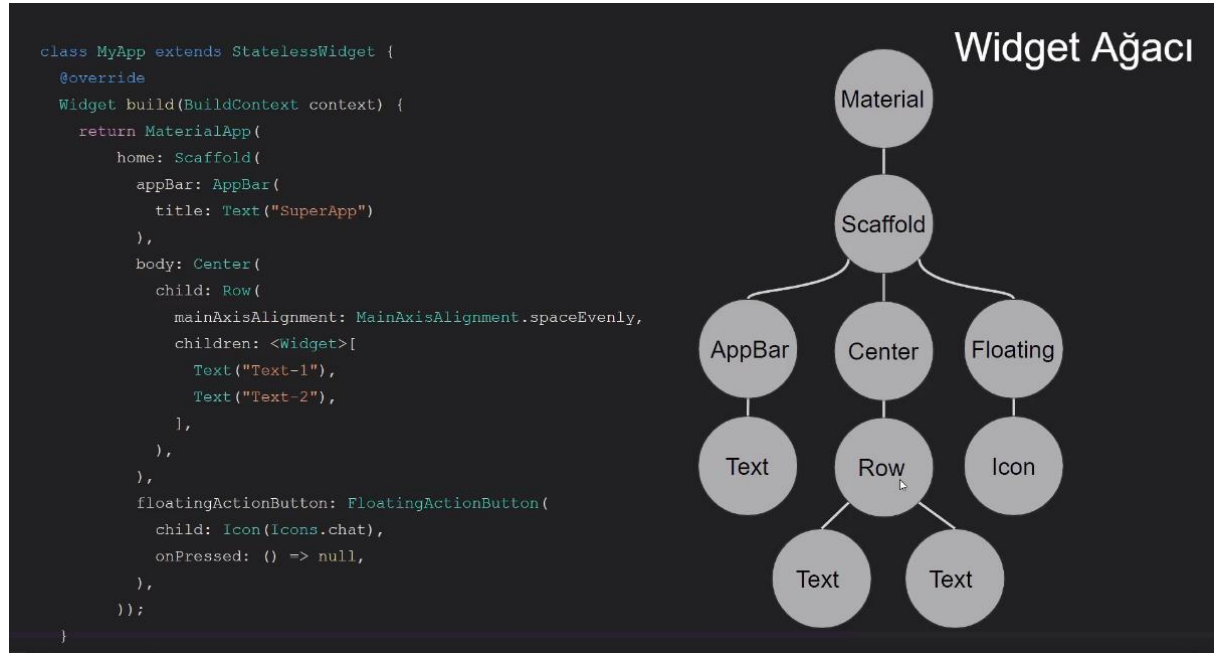
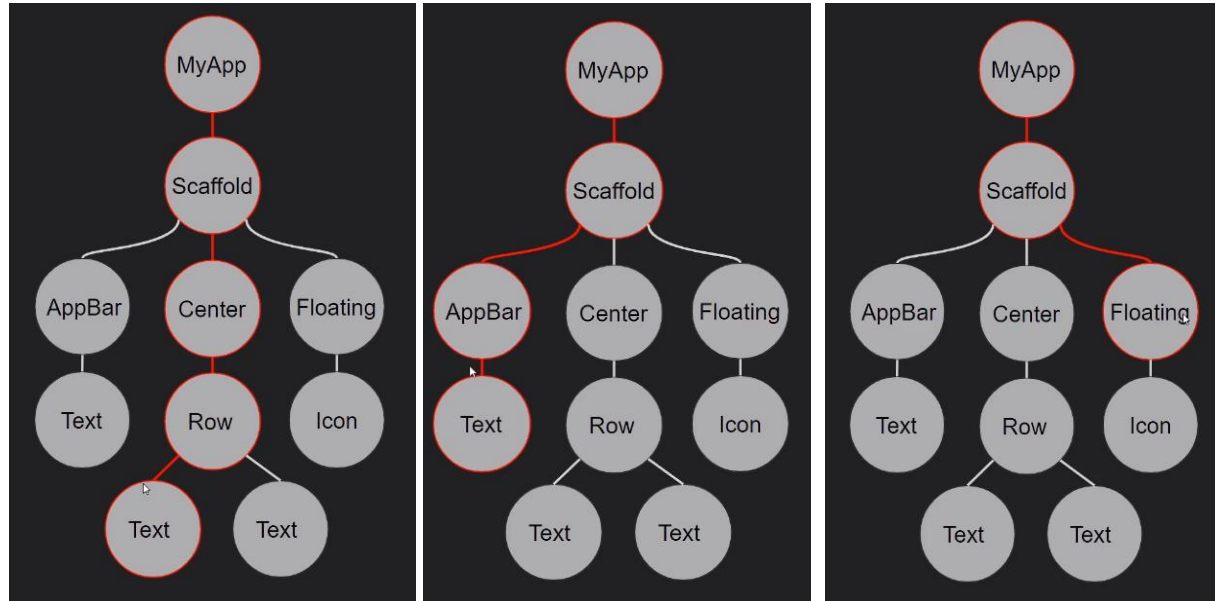


**BuildContext:** Widgetın widget ağacı içindeki konumudur. Her widgetın kendine ait bir BuildContexti bulunur. Üst widgetlarla iletişim kurmamızı sağlar.



BuildContext ile dikey konuma ulaşılır. Yataydaki widgetlarla ilgili bilgi vermez.



**build() Metdodu :** Build metodu BuildContext parametresi almak zorundadır. Çünkü build metodu yeni widgetları widget ağacına yerleştirir. Ama nereye yerleştirmesi gerektiğini bilmez. BuildContext parametresi, build metoduna, yeni widgetların nereye yerleşmesi gerektiğini gösteren bir haritadır. Build metodu gönderilen bu haritada önceki widgetların dikey olarak nasıl bir sırayla yerleştiklerini görür ve yeni widgetları sıranın sonuna ekler.

BuildContextteki contex kelimesi bağlam demektir. En üstten en alta birbirine bağlanmış widget topluluğuna bağlam denilebilir.

Bazen alt sıralardaki bir widgetın içinden üstteki bir widgetın özelliğini kullanmamız gerekebilir. Böyle durumlarda BuildContext üst widgetlarla iletişim kurmak için kullanılır.

**ElevatedButton()** : Klasik bir buton.

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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Butona bas'),
          onPressed: () {
            var alert=AlertDialog(
              title: Text("Mesaj Penceresi"),
              content: Text('Butona basıldı'),
            );
            showDialog(context: context, builder: (builder) => AlertDialog(
              title: Text("Mesaj Penceresi"),
              content: Text('Butona basıldı'),
            ));
          },
        ),
      ),
    ));
}
```

showDialogdaki builder parametresi bizden Buildercontext ile oluşturulan bir fonksiyon ister.

builder parametresi aşağıdaki gibide yazılabilir.

```
builder: (BuildContext context) {
  return alert;
}
```

**builder() widget:** build metodunun yaptığı gibi widgetları ağaca ekler. Builder widgetı sayesinde yeni bir widget sınıfı oluşturmaya gerek olmadan build işlemi yapılabilir.

Burada showDialogdaki context parametresi için gerekli olan context bilgisi bulunamaz.

Hem MaterialApp hemde MaterialApp metodunu çalıştıracak olan buton aynı build metodu içinde ağaca eklenirse o buton hiçbir zaman MaterialApp metodunu çalıştıramaz. Çünkü gönderilen build context içinde MaterialApp olmaz. MaterialApp'ın ağaca önceden eklenmiş olması gerekir. Sonuç olarak MaterialApp ve MaterialApp metodunu kullanacak olan buton farklı build metodlarıyla eklenmelidir.

Bunun için ayrı bir sınıf oluşturulması gerekir. Mesaj adında ayrı bir widget oluşturulur ve home parametresi mesaj widgetına yönlendirilir.

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

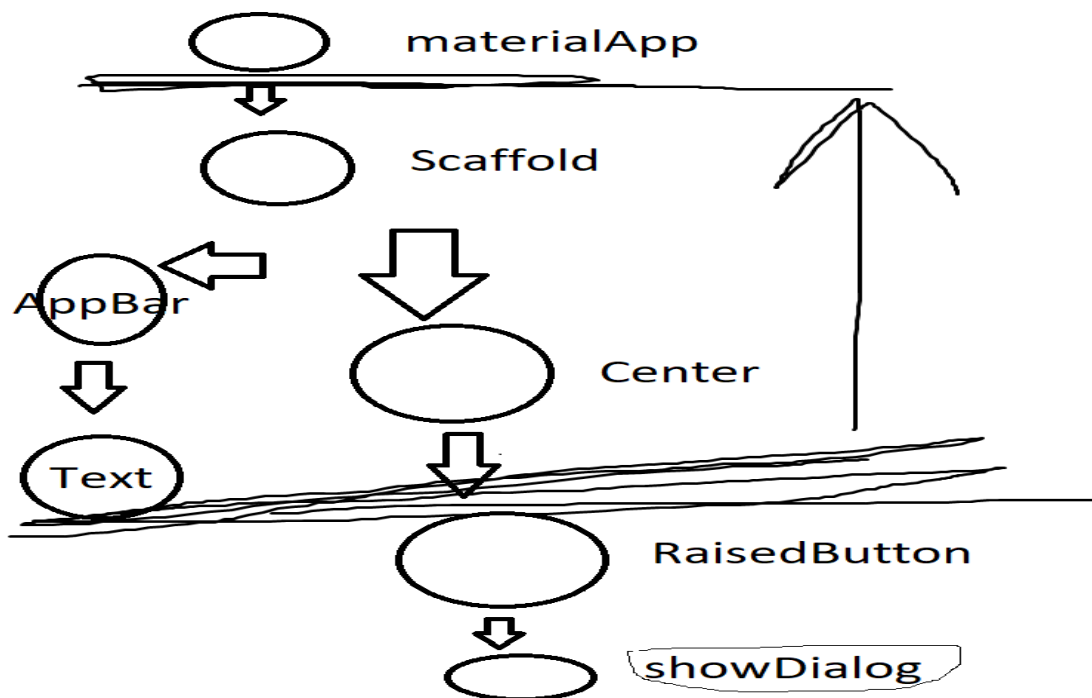
void main() {
  runApp(MaterialApp(
    home: mesaj()
  ));
}
```

```

class mesaj extends StatelessWidget {
  const mesaj({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Butona bas'),
          onPressed: () {
            var alert=AlertDialog(
              title: Text("Mesaj Penceresi"),
              content: Text('Butona basıldı'),
            );
            showDialog(context: context, builder: (builder) => AlertDialog(
              title: Text("Mesaj Penceresi"),
              content: Text('Butona basıldı'),
            ));
          },
        ),
      ),
    );
  }
}

```



Aşağıdaki, Yazı arka plan vurgusu olan kısım artık geçerli değil çünkü; Scaffold.of(context).showSnackBar metodu artık kullanımda değil.

Şimdi ekranın altında mesaj bilgisi verecek olan bir uygulama yapalım.

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

void main() {
  runApp(MaterialApp(home: mesaj()));
}

class mesaj extends StatelessWidget {
  const mesaj({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Butona bas'),
          onPressed: () {
            Scaffold.of(context)
              .showSnackBar(SnackBar(content: Text('Butona basıldı')));
          },
        ),
      ),
    );
  }
}
```

Çalıştırdığımız zaman hata verecek. Kırmızı yazılı hataya baktığımızda Scaffold.of metonunun context parametresine Scaffold context bilgisinin gelmediğini söylüyor.

Önceki örnekte olduğu gibi bu örnekte de Scaffold widgetının widget ağacına önceden eklenmesi gerekiyor.

Hem scaffold hem de scaffoldun metodunu çalıştıracak olan buton aynı build metodu içinde ağaca eklenirse o buton hiçbir zaman scaffoldun metodunu çalıştıramaz. Çünkü gönderilen build context içinde scaffold olmaz. Scaffold ağaca önceden eklenmiş olması gerekir. Sonuç olarak scaffold ve scaffoldu kullanacak olan buton farklı build metodlarıyla eklenmelidir. Butonu ayrı bir build içine almalıyız.

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

void main() {
  runApp(MaterialApp(home: mesaj()));
}

class mesaj extends StatelessWidget {
  const mesaj({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
```

```

        child: Buton(),
      ),
    );
  }
}

```

```

class Buton extends StatelessWidget {
  const Buton({
    Key? key,
  }) : super(key: key);

```

```

  @override
  Widget build(BuildContext context) {
    return ElevatedButton(
      child: Text('Butona bas'),
      onPressed: () {
        Scaffold.of(context)
          .showSnackBar(SnackBar(content: Text('Butona basıldı')));
      },
    );
  }
}

```

Biraz önce Buton isminde yeni bir widget oluşturmak zorunda kaldık. Builder widgetı ile yeni bir sınıf oluşturmadan butonu widget ağacına ekleyebilir ve widget ağacına eklenmiş olan Scaffold widgetını görmesini sağlayabiliriz.

Bunun için child: parametresinden Buton sınıfını silip butonu buraya builder widgetı ile ekleyelim.

```

import 'package:flutter/material.dart';

```

```

void main() {
  runApp(MaterialApp(home: mesaj()));
}

```

```

class mesaj extends StatelessWidget {
  const mesaj({Key? key}) : super(key: key);

```

```

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
        child: Builder(builder: (BuildContext context) {
          return ElevatedButton(
            child: Text('Butona bas'),
            onPressed: () {
              Scaffold.of(context)
                .showSnackBar(SnackBar(content: Text('Butona basıldı')));
            },
          ),
        }),
      ),
    );
  }
}

```

Yukarıdaki Yazı arka plan vurgusu olan kısım artık geçerli değil

**Scaffold.of yerine ScaffoldMessenger.of :**Bu metodla contexte Scaffold bilgisini göndermemize gerek kalmaz. Yani Builder kullanmadan Scaffold altında butonu direkt kullanabiliriz.

```
ScaffoldMessenger.of(context)
  .showSnackBar(SnackBar(content: Text('Butona basıldı')));

import 'package:flutter/material.dart';

void main() {
  runApp(MaterialApp(home: mesaj()));
}

class mesaj extends StatelessWidget {
  const mesaj({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Mobil'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Butona bas'),
          onPressed: () {
            ScaffoldMessenger.of(context)
              .showSnackBar(SnackBar(content: Text('Butona basıldı')));

            // Scaffold.of(context)
            //   .showSnackBar(SnackBar(content: Text('Butona basıldı')));
          },
        ),
      ),
    );
  }
}
```

## RESİM EKLEME

Projemizde en üstte Project tabında sağ tıklayıp yeni klasör açalım ve ismine images verelim.

Kullanacağımız resimleri bu klasör içerisine alıyoruz.

Bunları direk kullanamayız. Bunu fluttera tanıtmak gerekir. Bunuda pubspec.yaml dosyasının içerisinde yaparız.

Pubspec dosyasını açıp assets kısmına geliriz. Assets bölümü uygulamaya ekleyeceğimiz varlıklar kısmıdır.

Assets bölümünü aktif hale getiririz. Alt satırdaki koduda aktif hale getiririz. Resim dosyası ismini sileriz.

```
assets:
  - images/
```

Kaydedip pub get butonuna basarak geçerli hale getiririz.

Sonra Image widgetını kullanırız.

```
Image.asset('images/resim.jpg'),
```

## UYGULAMAYA İKON VERME

Herhangi bir resim dosyası, icon yapan online bir sayfadan icon olarak oluşturulur.

Örnek : <https://appicon.co/>

Oluşan icon dosyasındaki aşağıda ismi geçen klasörleri ilgili yerlere kopyala.

ANDROID : C:\Users\DERS\StudioProjects\untitled3\android\app\src\main\res 5 adet mipmap klasörünü değiştir.

IOS : C:\Users\DERS\StudioProjects\untitled3\ios\Runner Assets.xcassets klasörünü değiştir.

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

void main() {
  runApp(MaterialApp(
    debugShowCheckedModeBanner: false,
    title: 'Week 4',
    home: Scaffold(
      appBar: AppBar(
        title: Text(
          'Başlık',
          style: TextStyle(
            fontSize: 20,
            color: Colors.black,
            //backgroundColor: Colors.blue,
            //fontWeight: FontWeight.bold,
          ),
        ),
        backgroundColor: Colors.green,
        centerTitle: true,
      ),
      body: Center(
        child: Column(
          //mainAxisAlignment: MainAxisAlignment.spaceAround,
          children: [
            Text(
              'Ana Sayfa',
              style: TextStyle(
                fontSize: 20,
                color: Colors.orange,
                backgroundColor: Colors.blue,
                //fontWeight: FontWeight.bold,
              ),
            ),
          ],
        ),
      ),
    ),
  );
}
```

```

    ),
    SizedBox(height: 50,),
    Row(
      mainAxisAlignment: MainAxisAlignment.center,
      children: [
        RaisedButton(onPressed: (){},child: Text("Buton"),),
        SizedBox(width: 50,),
      ],
    ),
    Image.asset('images/resim.jpg'),],
  ),
),
floatingActionButton: FloatingActionButton(
  onPressed: () {
    print("Tıklandı");
  },
  backgroundColor: Color(0xffC3D6B7),
),
drawer: Drawer(),
//bottomNavigationBar: NavigationToolbar(),
),
));
}

```

Aşağıdaki kullanımları öğren.

SizedBox() ?????

Color(0xFF???) ????