

Flutter ile Uygulama Geliştirme Kursu | Android & IOS

Flutter Tasarım

Kasım ADALAN

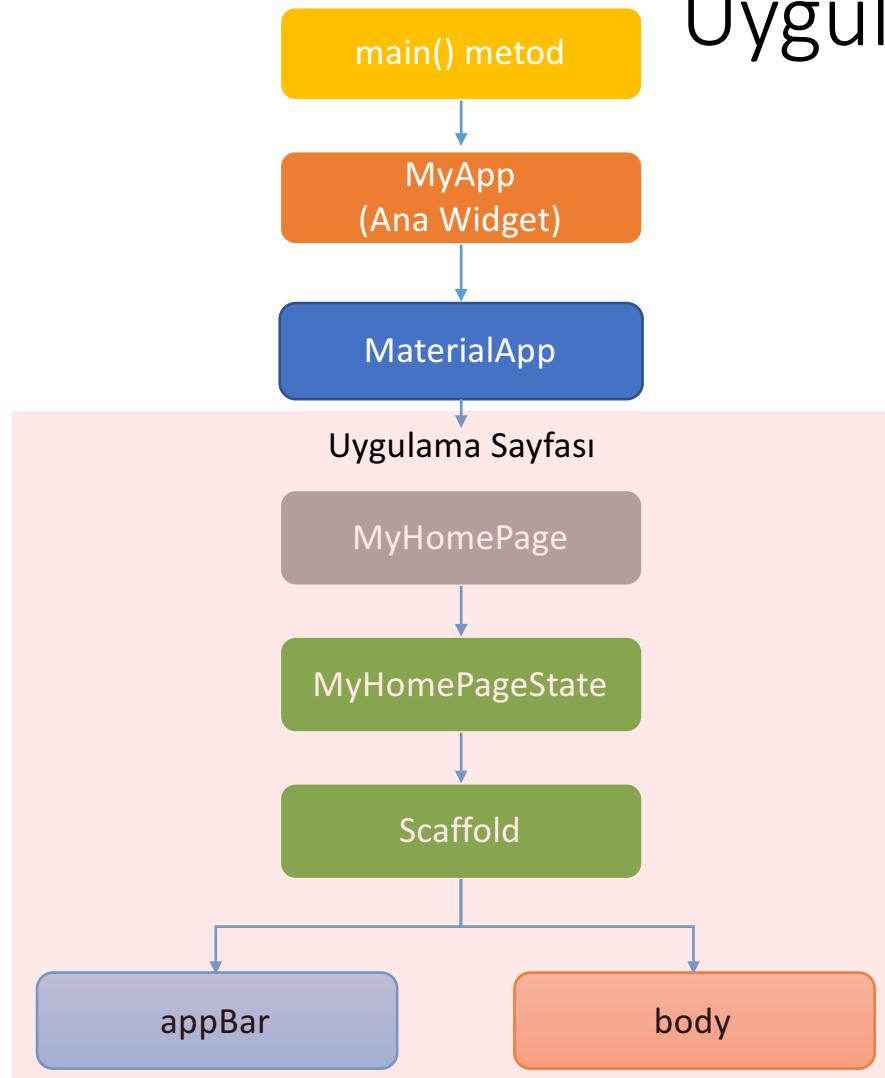
Elektronik ve Haberleşme Mühendisi
Android - IOS Developer and Trainer

Eğitim İçeriği

- Tasarım Giriş
- Row – Column –Stack
- MainAxisAlignment
- CrossAxisAlignment
- MainAxisAlignment
- Center
- Align
- SizedBox
- Spacer
- Padding
- Expanded
- Özelleştirilmiş Widget
- Çoklu Ekran Desteği
- Icon Oluşturma

Tasarım Giriş

Uygulama Yapısı



Scaffold (Sayfanın Tamamı)



Kasım ADALAN

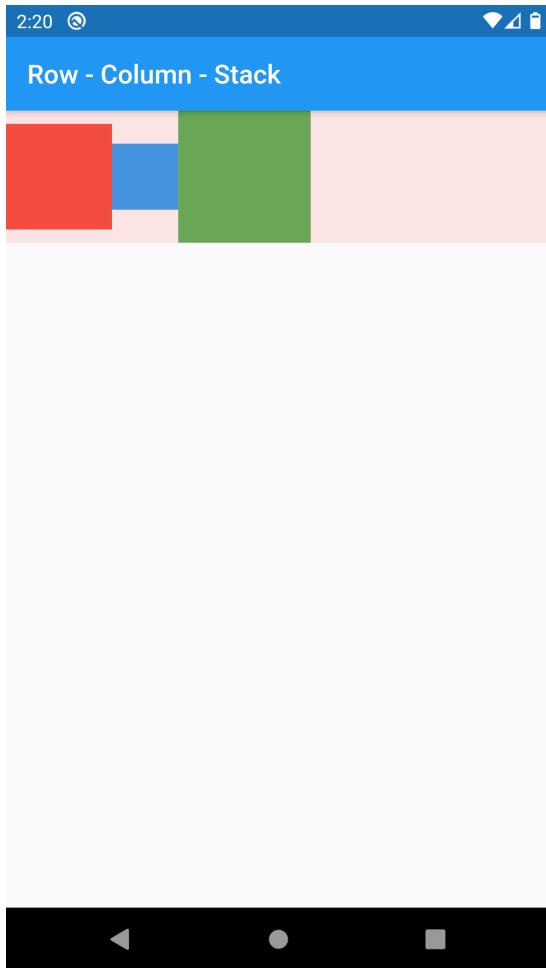
Tasarım Sayfa Yapısı

```
class _MyHomePageState extends State<MyHomePage> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Column(  
        children: <Widget>[  
          Text("Merhaba"),  
        ], // <Widget>[]  
      ), // Column  
    ); // Scaffold  
  }  
}
```

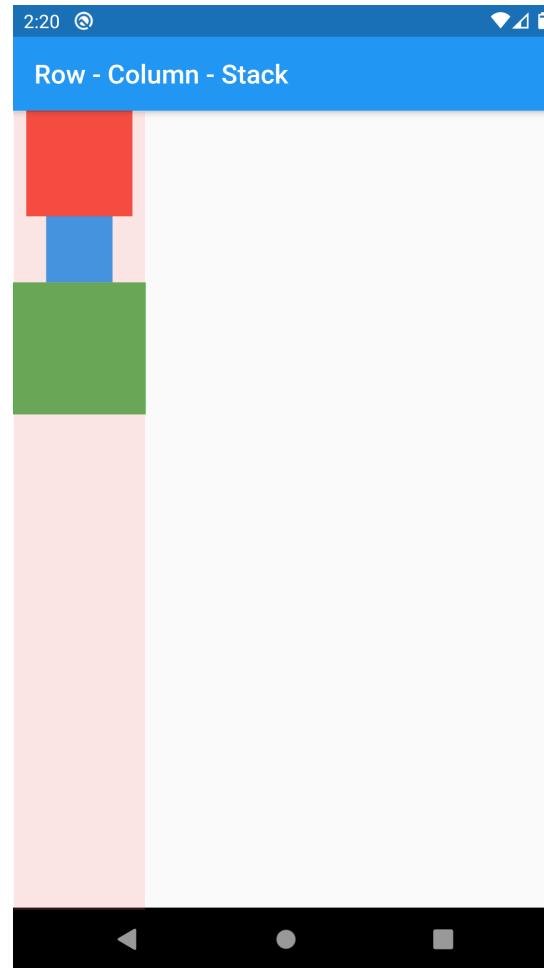


Row – Column - Stack

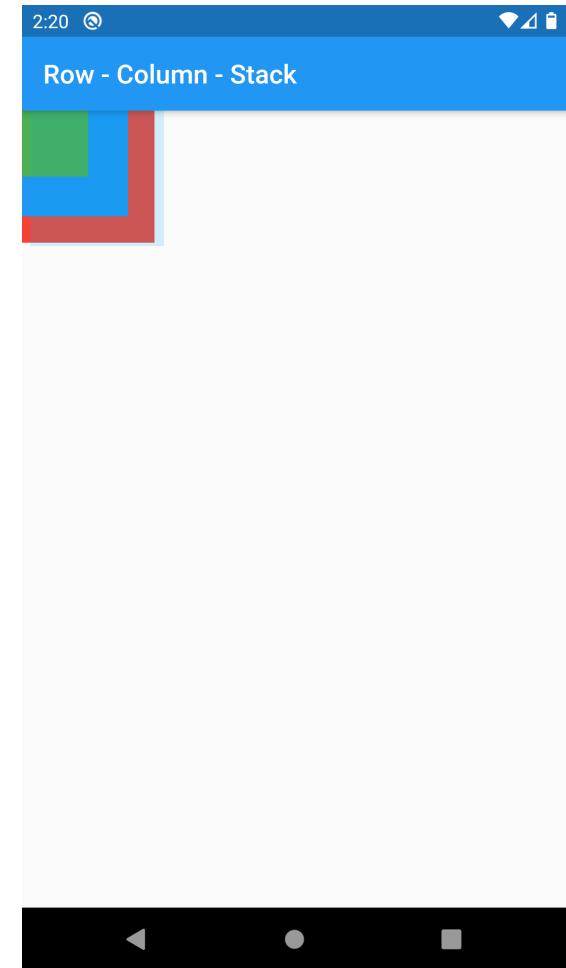
Row



Column



Stack

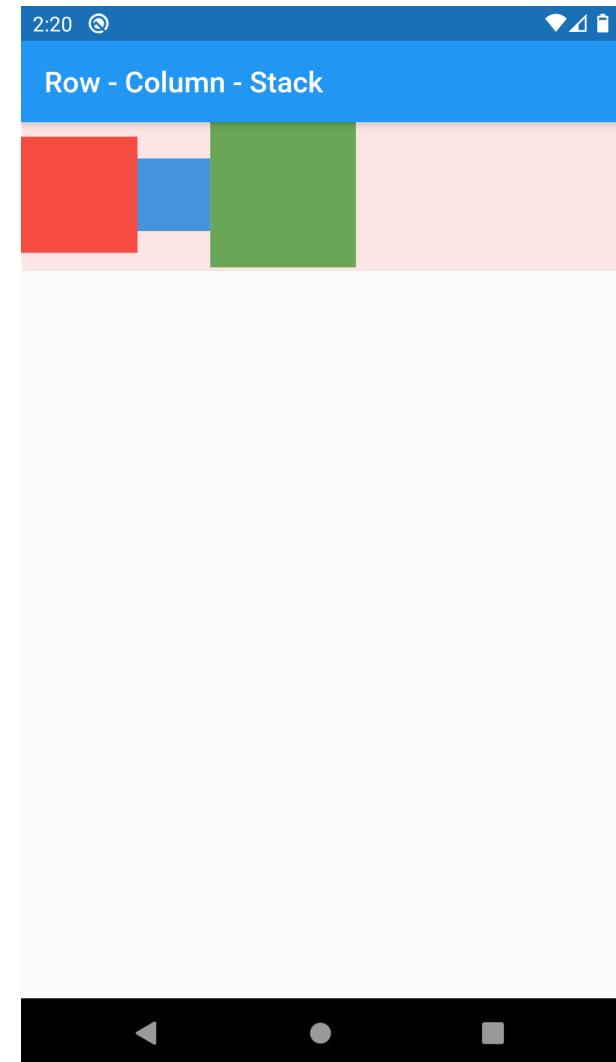


Row

Row

- Widgetların yan yana durmasını sağlar.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.

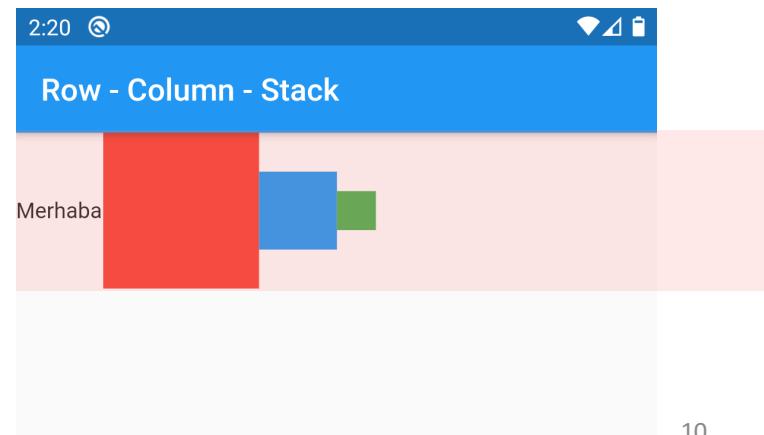
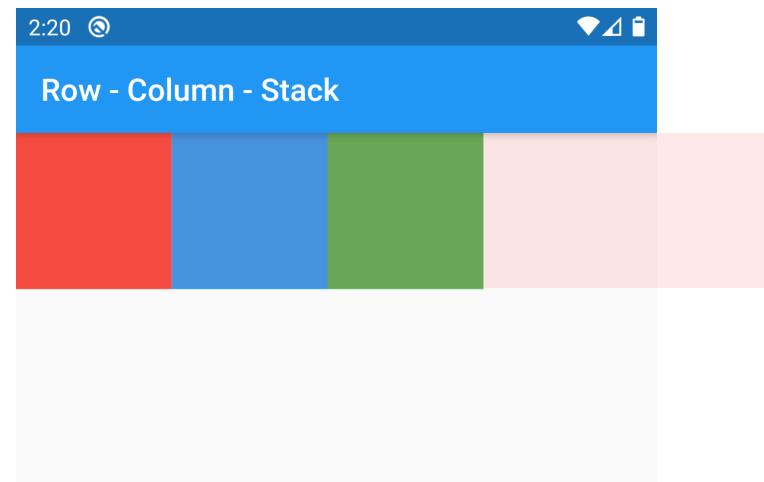
```
class _MyHomePageState extends State<MyHomePage> {  
    @override  
    Widget build(BuildContext context) {  
        return Scaffold(  
            appBar: AppBar(  
                title: Text(widget.title),  
            ), // AppBar  
            body: Row(  
                children: <Widget>[  
                    Container(width: 80,height: 80,color : Colors.red),  
                    Container(width: 50,height: 50,color : Colors.blue),  
                    Container(width: 100,height: 100,color : Colors.green),  
                ], // <Widget>[]  
            ), // Row  
        ); // Scaffold  
    }  
}
```



Row

```
body: Row(  
    children: <Widget>[  
        Container(width: 100,height: 100,color : Colors.red),  
        Container(width: 100,height: 100,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
, // Row
```

```
body: Row(  
    children: <Widget>[  
        Text("Merhaba"),  
        Container(width: 100,height: 100,color : Colors.red),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 25,height: 25,color : Colors.green),  
    ], // <Widget>[]  
, // Row
```

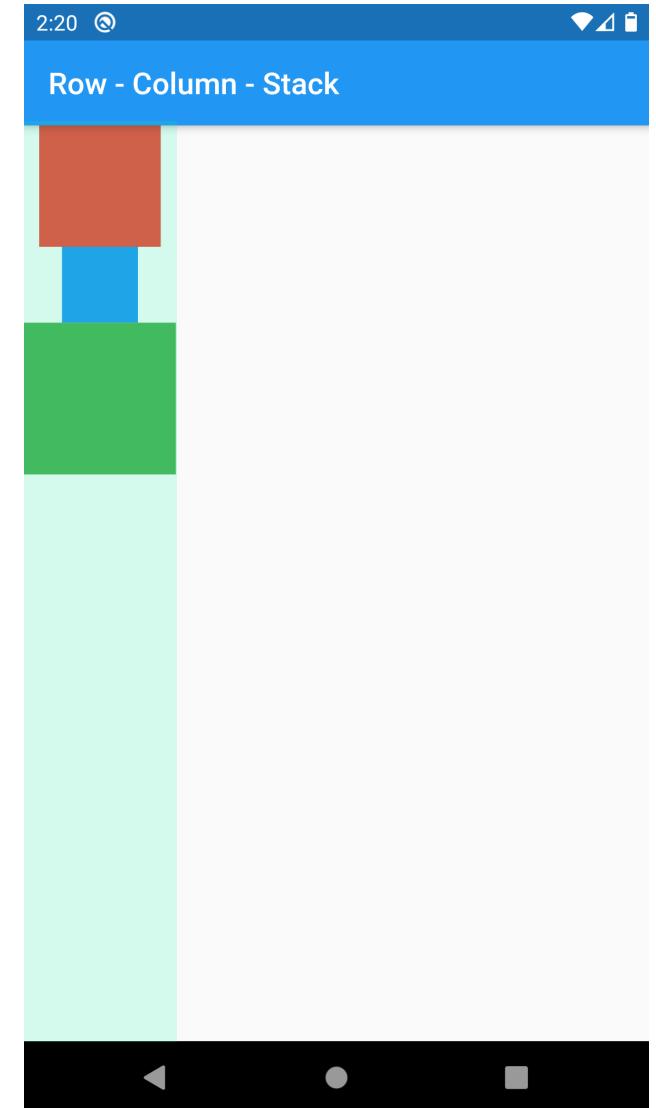


Column

Column

- Widgetların alt alta durmasını sağlar.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.

```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Column(  
        children: <Widget>[  
          Container(width: 80,height: 80,color : Colors.red),  
          Container(width: 50,height: 50,color : Colors.blue),  
          Container(width: 100,height: 100,color : Colors.green),  
        ], // <Widget>[]  
      ), // Column  
    ); // Scaffold  
  }  
}
```



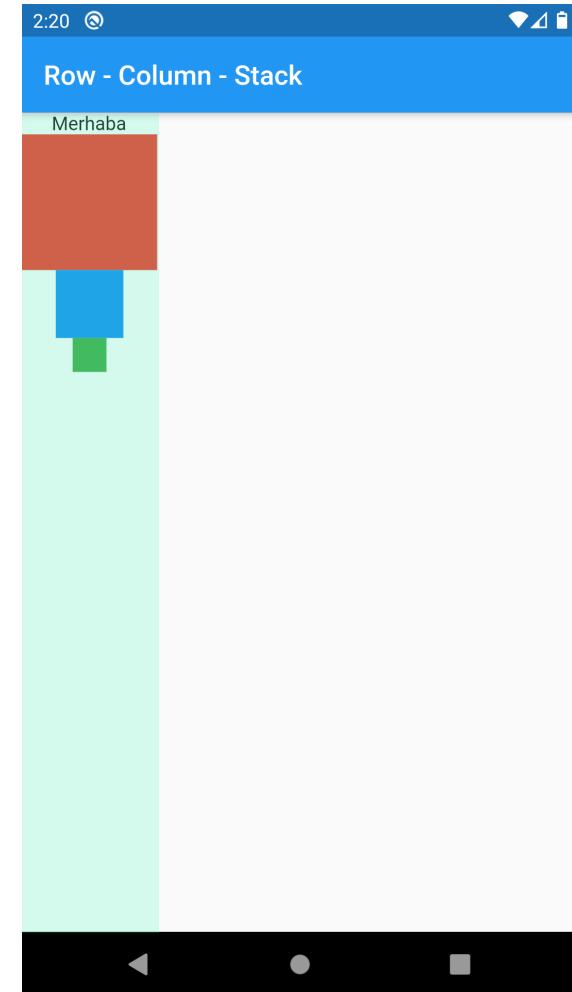
Column

```
body: Column(  
    children: <Widget>[  
        Container(width: 100,height: 100,color : Colors.red),  
        Container(width: 100,height: 100,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
, // Column
```



Column

```
body: Column(  
    children: <Widget>[  
        Text("Merhaba"),  
        Container(width: 100,height: 100,color : Colors.red),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 25,height: 25,color : Colors.green),  
    ], // <Widget>[]  
, // Column
```



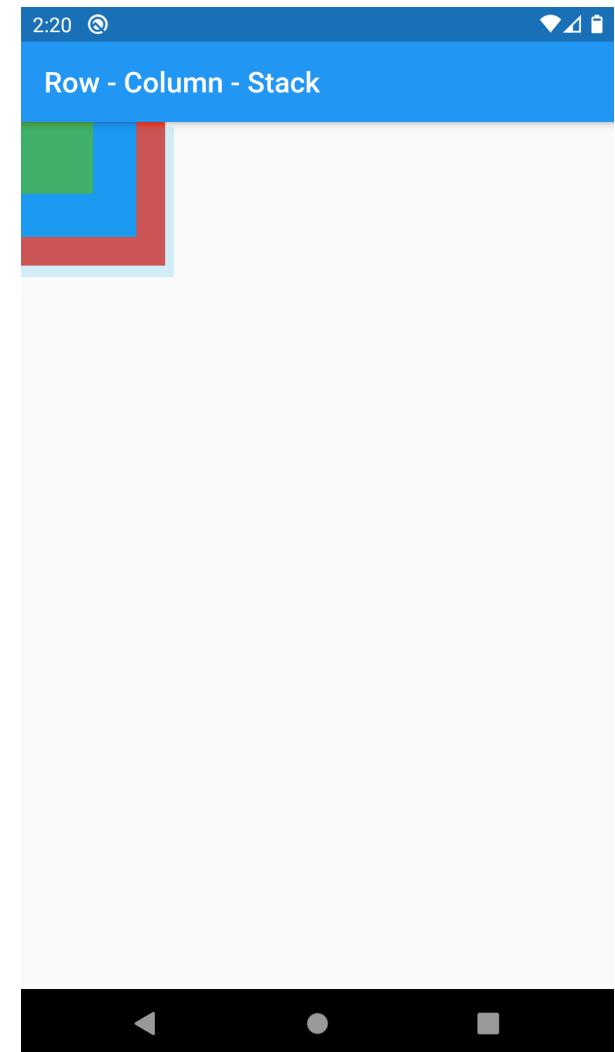
Stack

Stack

- Widgetların üst üste durmasını sağlar.
- İlk eklenen en altta yer alır.
- Varsayılan başlangıç noktası sol üst köşedir.
- Boyutu içindeki widgetlara göre değişir.

```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Stack(  
        children: <Widget>[  
          Container(width: 100,height: 100,color : Colors.red),  
          Container(width: 80,height: 80,color : Colors.blue),  
          Container(width: 50,height: 50,color : Colors.green),  
        ], // <Widget>[]  
      ), // Stack  
    ); // Scaffold  
  }  
}
```

Nasim ADALAN

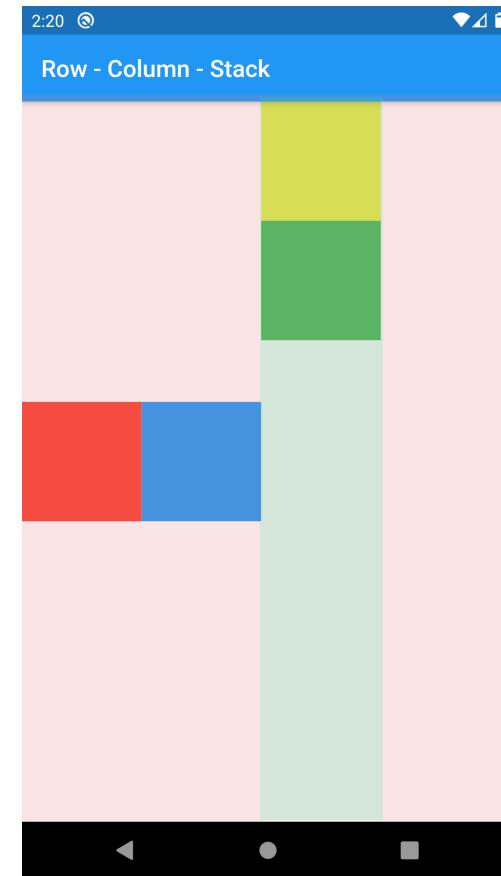


Row & Column

```
body: Row(  
    children: <Widget>[  
        Container(width: 100,height: 100,color : Colors.red),  
        Container(width: 100,height: 100,color : Colors.blue),  
        Column(  
            children: <Widget>[  
                Container(width: 100,height: 100,color : Colors.yellow),  
                Container(width: 100,height: 100,color : Colors.green),  
            ], // <Widget>[]  
        ), // Column  
    ], // <Widget>[]  
>, // Row
```

Row içindeki widget'ların ortada durmalarının sebebi row'un içine column eklendi ve ekranın dikeyinde bir alan oluşturdu bu alan row içindekileri dikeyde ortaladı.

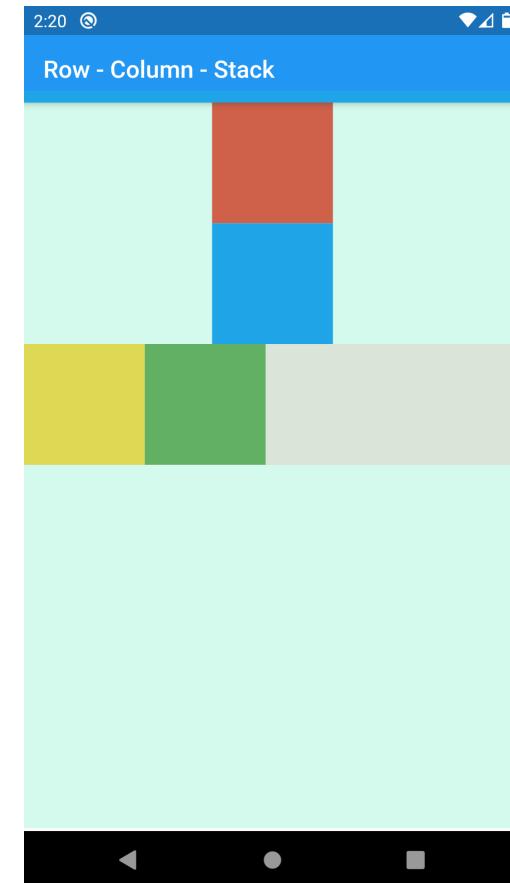
Kasım ADALAN



Row & Column

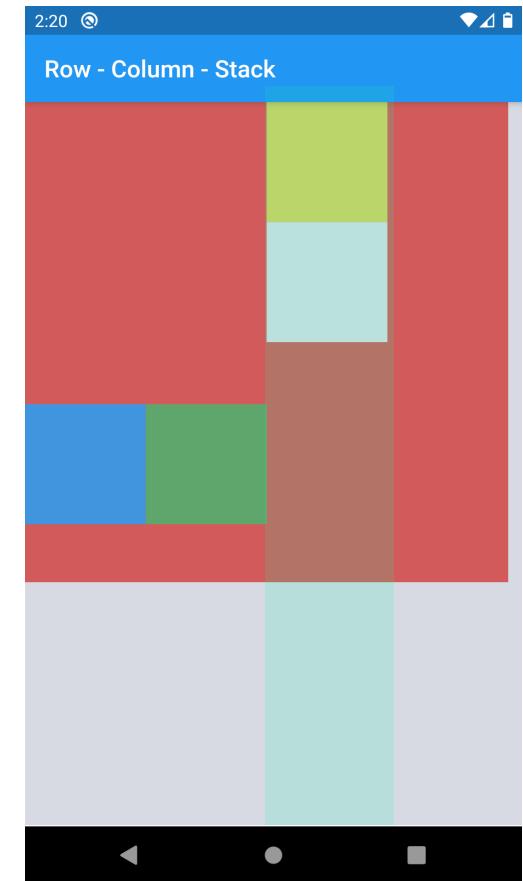
```
body: Column(  
  children: <Widget>[  
    Container(width: 100,height: 100,color : Colors.red),  
    Container(width: 100,height: 100,color : Colors.blue),  
    Row(  
      children: <Widget>[  
        Container(width: 100,height: 100,color : Colors.yellow),  
        Container(width: 100,height: 100,color : Colors.green),  
      ], // <Widget>[]  
    ), // Row  
  ], // <Widget>[]  
, // Column
```

Column içindeki widget'ların ortada durmalarının sebebi column'un içine row eklendi ve ekranın yatayında bir alan oluşturdu bu alan column içindekileri yatayda ortaladı.



Row – Column - Stack

```
body: Stack(  
    children: <Widget>[  
        Container(width: 400,height: 400,color : Colors.red),  
        Row(  
            children: <Widget>[  
                Container(width: 100,height: 100,color : Colors.blue),  
                Container(width: 100,height: 100,color : Colors.green),  
                Column(  
                    children: <Widget>[  
                        Container(width: 100,height: 100,color : Colors.yellow),  
                        Container(width: 100,height: 100,color : Colors.white),  
                    ], // <Widget>[]  
                ), // Column  
            ], // <Widget>[]  
        ), // Row  
  
, // <Widget>[]  
>, // Stack
```

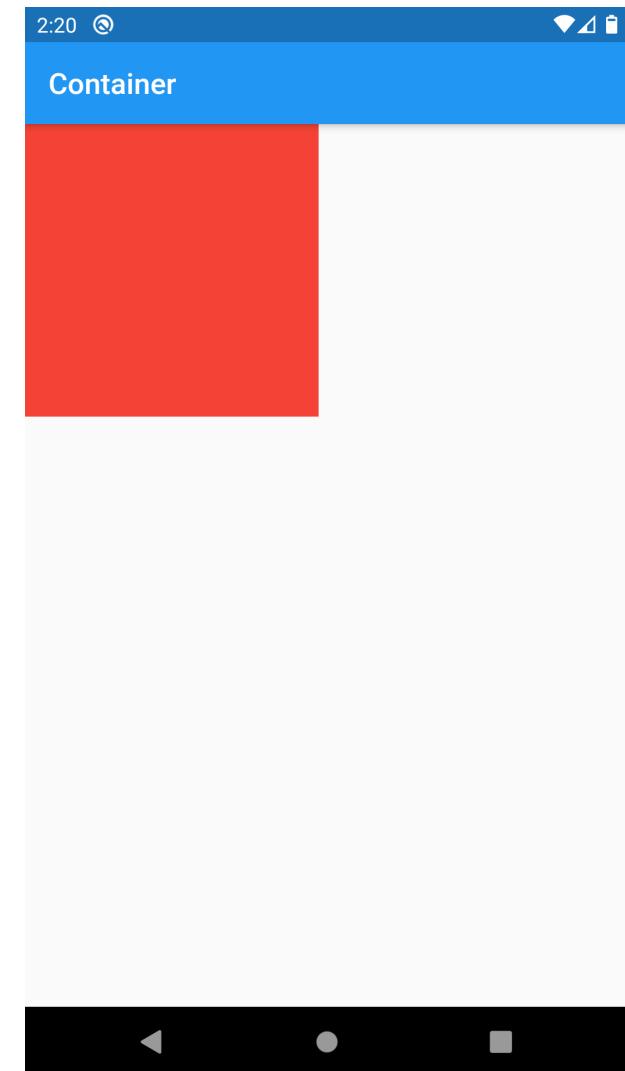


Container

Container

- Görsel nesneler için alan oluşturmak ve bu alan ile görsel nesneye özellikler vermek için kullanılır.

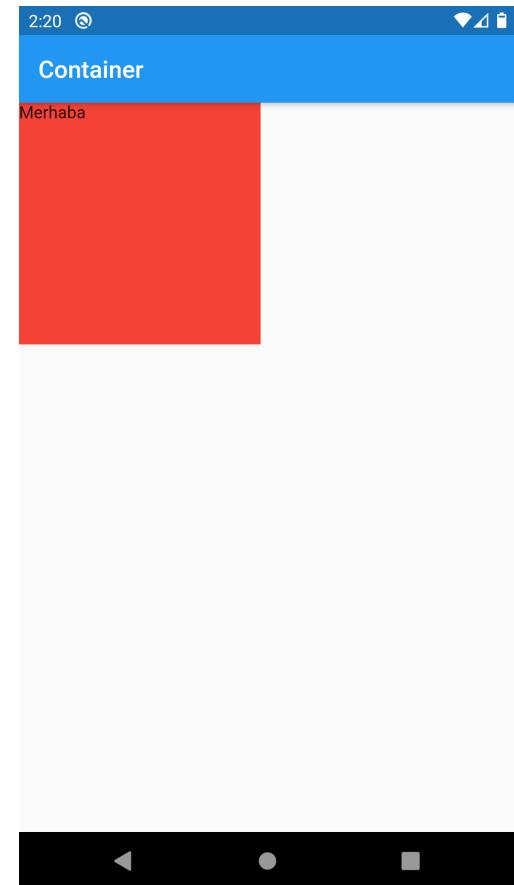
```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Row(  
        children: <Widget>[  
          Container(  
            width: 200, //Genişlik  
            height: 200, //Yükseklik  
            color: Colors.red, //Renk  
          ), // Container  
        ], // <Widget>[]  
      ), // Row  
    ); // Scaffold  
  }  
}
```



Kasım ADALAN

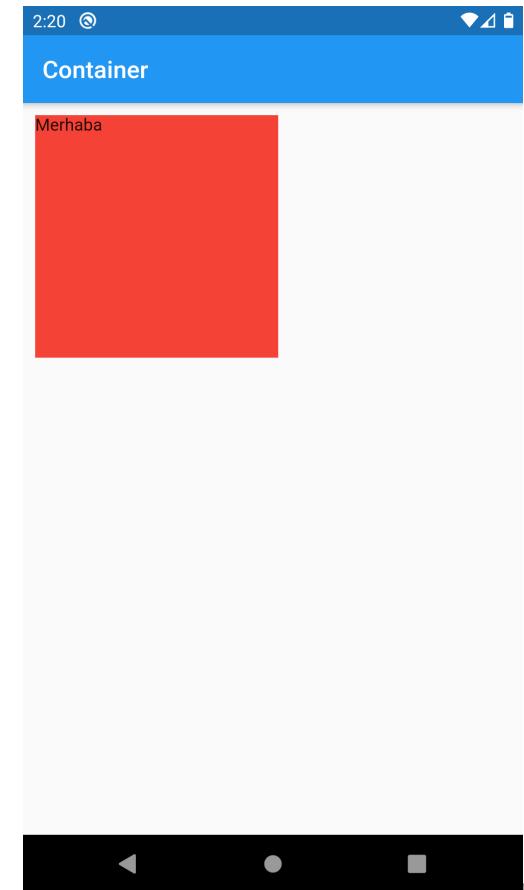
Text Ekleyebiliriz.

```
body: Row(  
    children: <Widget>[  
        Container(  
            width: 200, //Genişlik  
            height: 200, //Yükseklik  
            color: Colors.red, //Renk  
            child: Text("Merhaba"),  
        ), // Container  
    ], // <Widget>[]  
, // Row
```



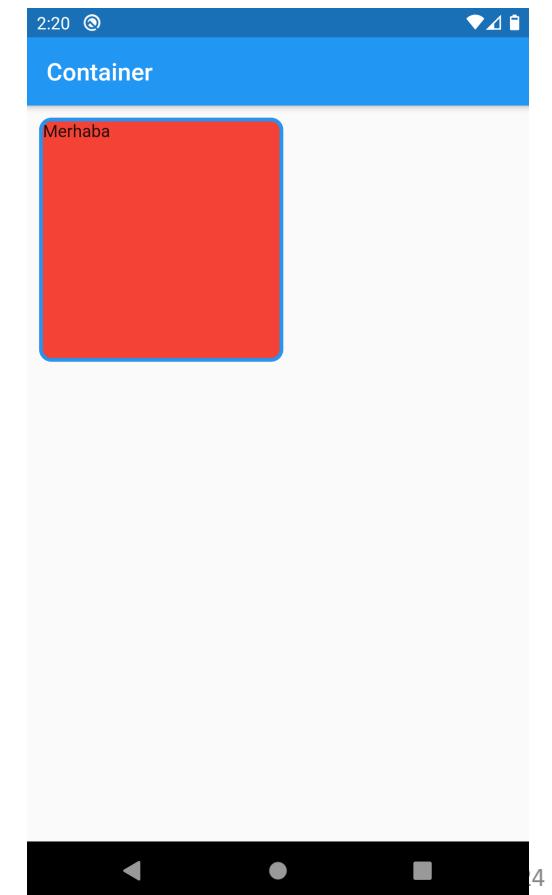
Çevresine Boşluk Ekleyebiliriz.

```
body: Row(  
    children: <Widget>[  
        Container(  
            margin: const EdgeInsets.all(10.0),  
            width: 200, //Genişlik  
            height: 200, //Yükseklik  
            color: Colors.red, //Renk  
            child: Text("Merhaba"),  
        ), // Container  
    ], // <Widget>[]  
, // Row
```



Container Özelleştirme

```
body: Row(  
  children: <Widget>[  
    Container(  
      margin: const EdgeInsets.all(10.0),  
      width: 200, //Genişlik  
      height: 200, //Yükseklik  
      child: Text("Merhaba"),  
      decoration: BoxDecoration(  
        color: Colors.red, //Arkaplan Renk  
        border: Border.all(  
          color: Colors.blue, //Çerçeve Renk  
          width: 3.0, //Çerçeve Kalınlık  
        ), // Border.all  
        //Köselere oval  
        borderRadius: BorderRadius.all(Radius.circular(10.0)),  
      ), // BoxDecoration  
    ), // Container  
  ], // <Widget>[]  
) // Row
```



Tasarım Alanı Kodlama Mantığı

Tasarım Alanı Kodlama Mantığı

- Flutter içinde hersey widget olarak tanımlanır.Tasarım yapılarının genel ismi widget'tır.
- Kodlamalar iç içe bir yapı ile ilerlemektedir.
- Eğer Row veya Column altına birden fazla widget eklenecekse **children: <Widget>[** yapısı ile eklenir.
- **children: <Widget>[** altında child ile başlanmaz direk widget ismi yazılır.**children: <Widget>[** dizidir ve içindeki child olacağı için child ile belirtilmesine gerek yoktur.
- **children: <Widget>[** altına widget eklenmiş ise ve bu widget içinede başka bir widget eklenecekse **child** ile başlanır.

```
  body: Center(  
    child: Row(  
      children: <Widget>[  
        Container(  
          color: Colors.red,  
          width: 300,  
          height: 300,  
          child: Center(  
            child: Text("Merhaba")  
          ), // Center  
        ), // Container  
      ], // <Widget>[]  
    ), // Row  
  ), // Center
```

Row & Column

İçeriği Hizalama İşlemleri

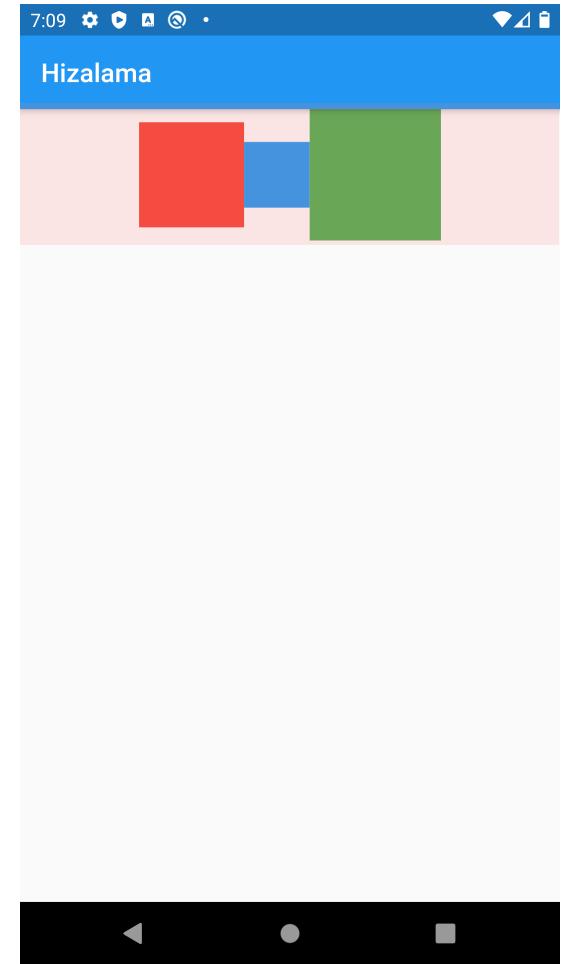
```
# MainAxisAlignment  
# CrossAxisAlignment  
# MainAxisSize
```

MainAxisAlignment

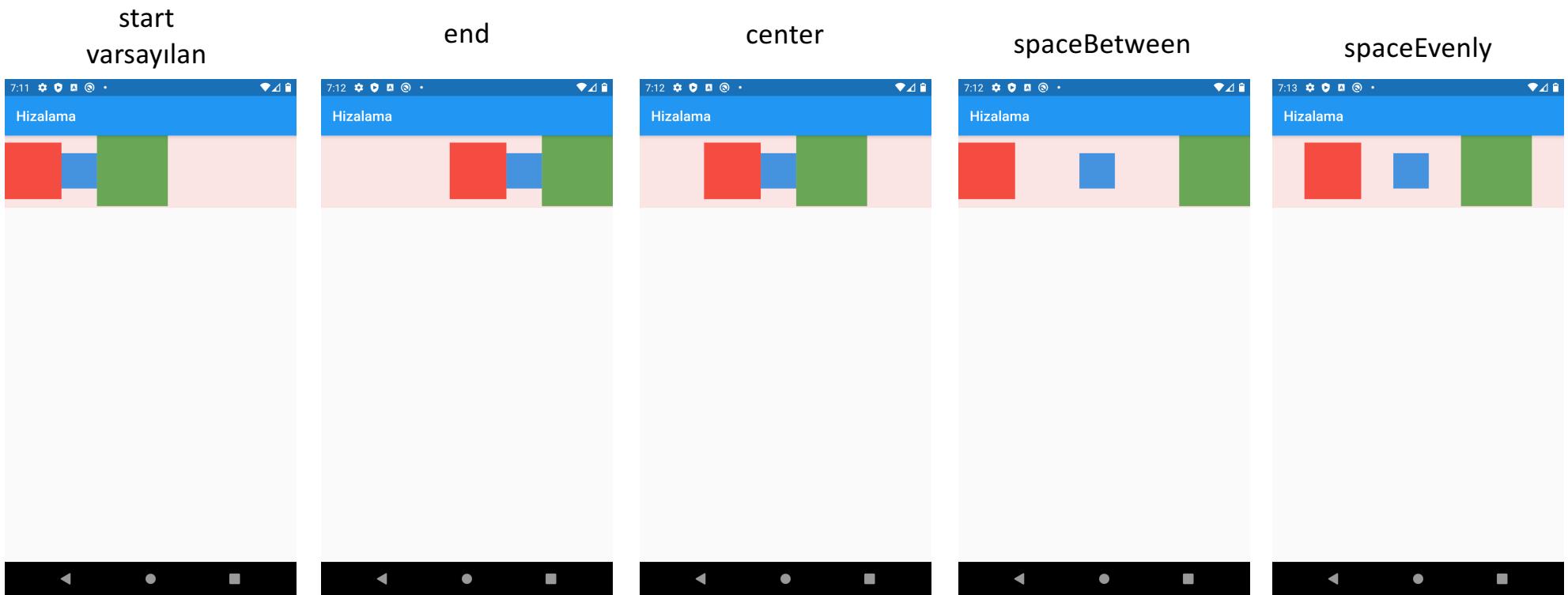
MainAxisAlignment özelliği

- Row veya Column **içinde** Row ise yatayda , column ise dikeyde hizalama yapılır.

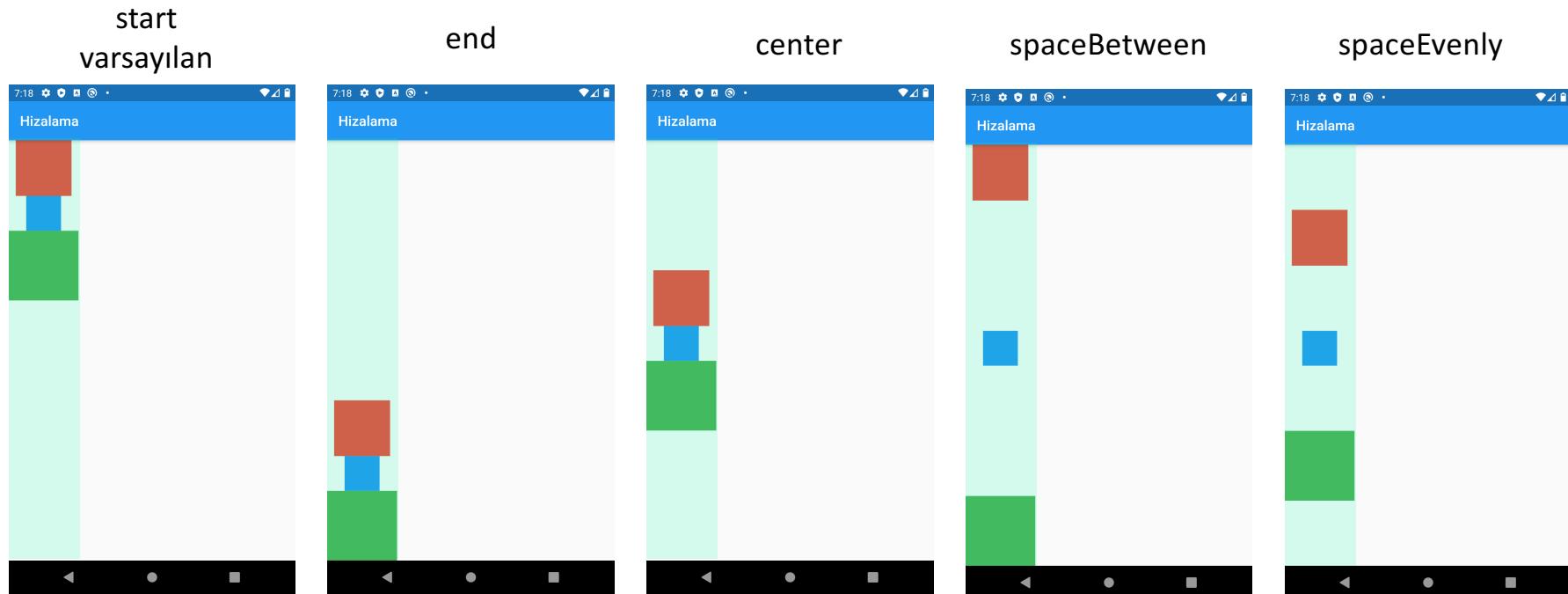
```
body: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```



MainAxisAlignment - Row



MainAxisAlignment - Column

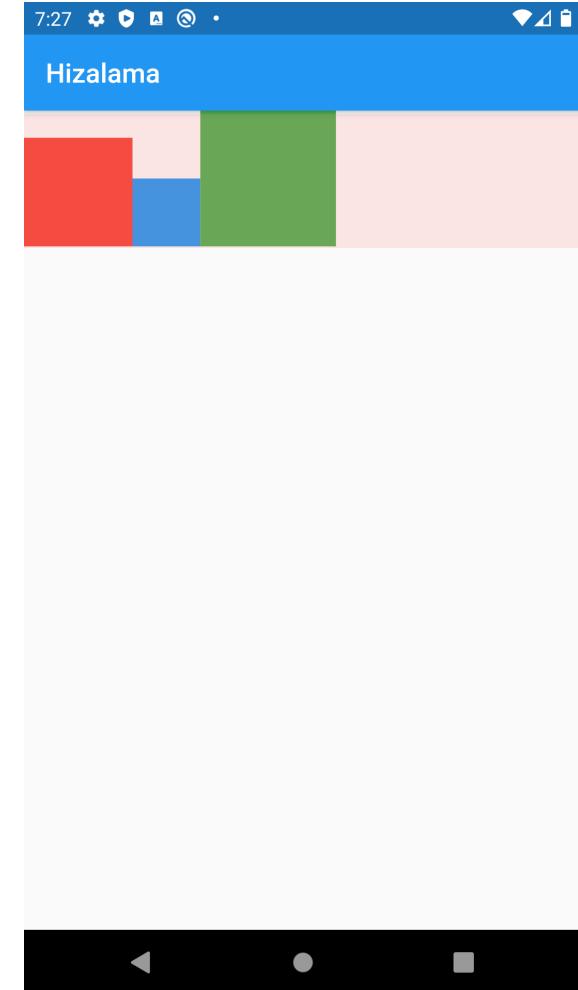


CrossAxisAlignment

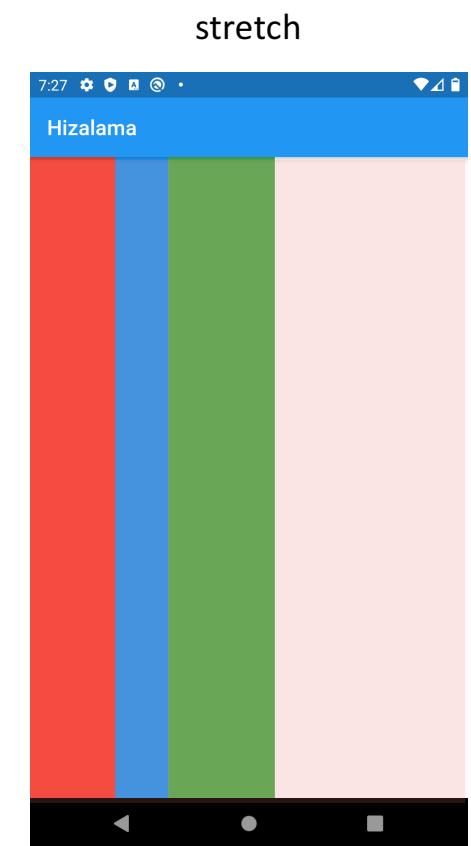
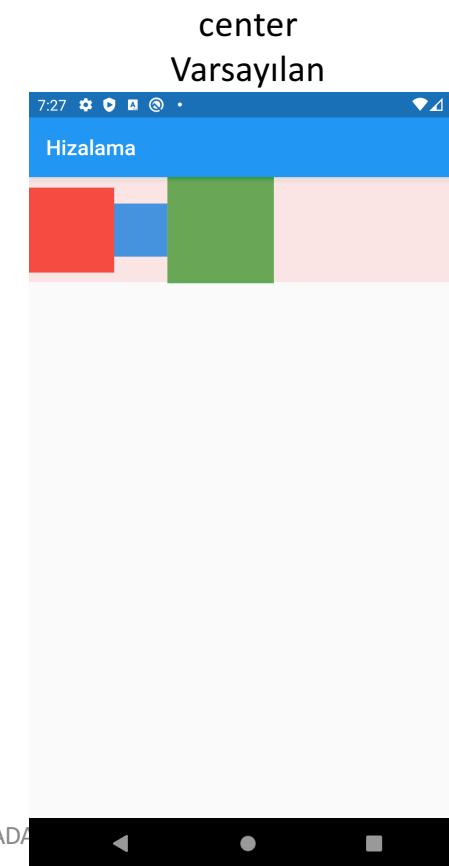
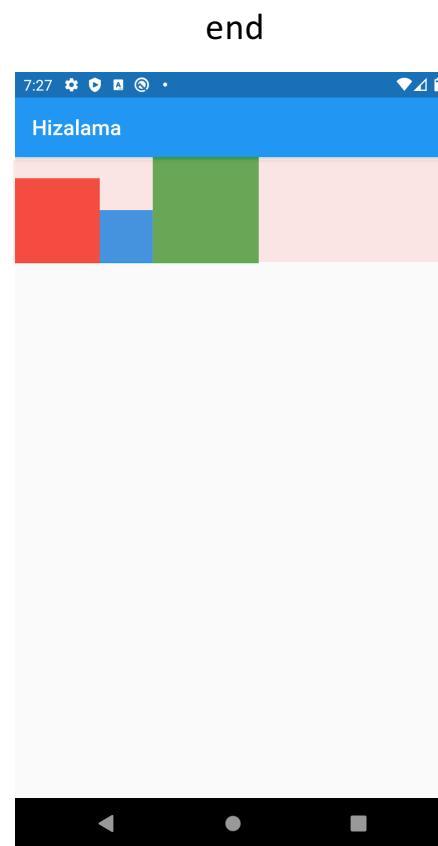
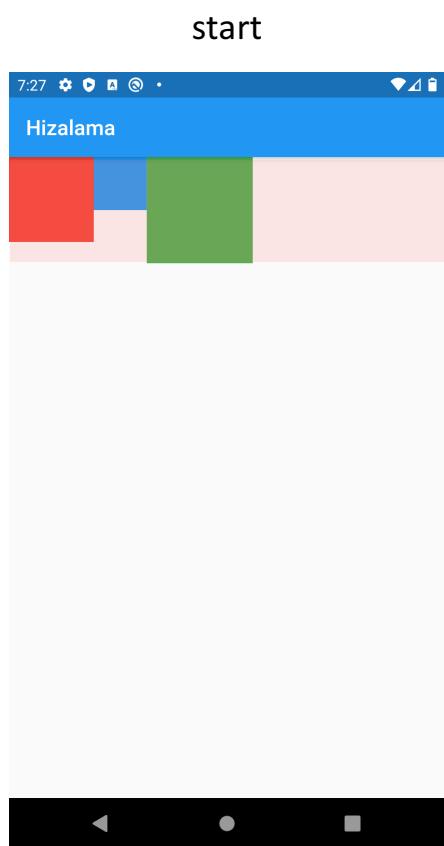
CrossAxisAlignment özelliği

- MainAxisAlignment'in tersi yönde hizalama yapar.
- Row veya Column **içinde** satırsa dikeyde, column ise yatayda hizalama yapılır.
- Bunu sağlamak için row veya column **içinde** hareket alanı olmalıdır.

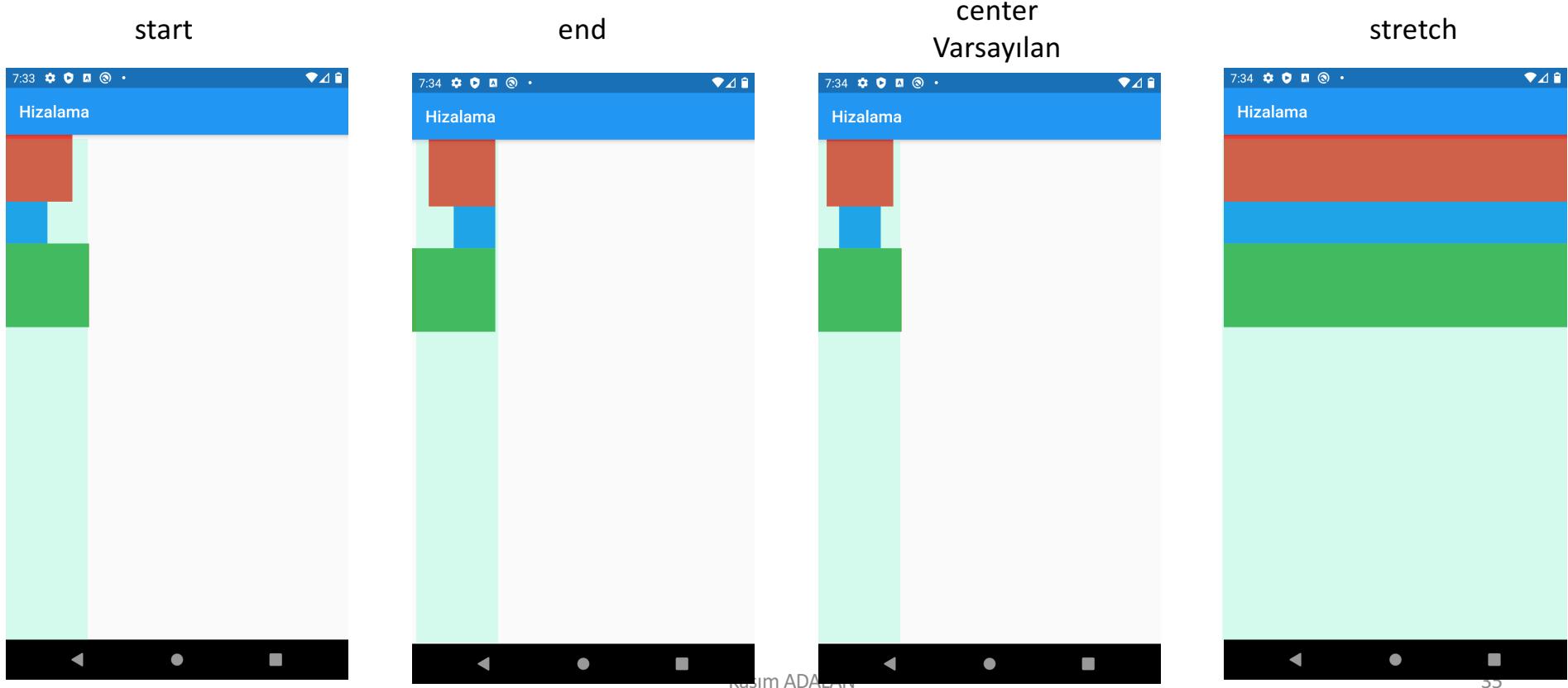
```
body: Row(  
    crossAxisAlignment: CrossAxisAlignment.end,  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```



CrossAxisAlignment - Row



CrossAxisAlignment - Column

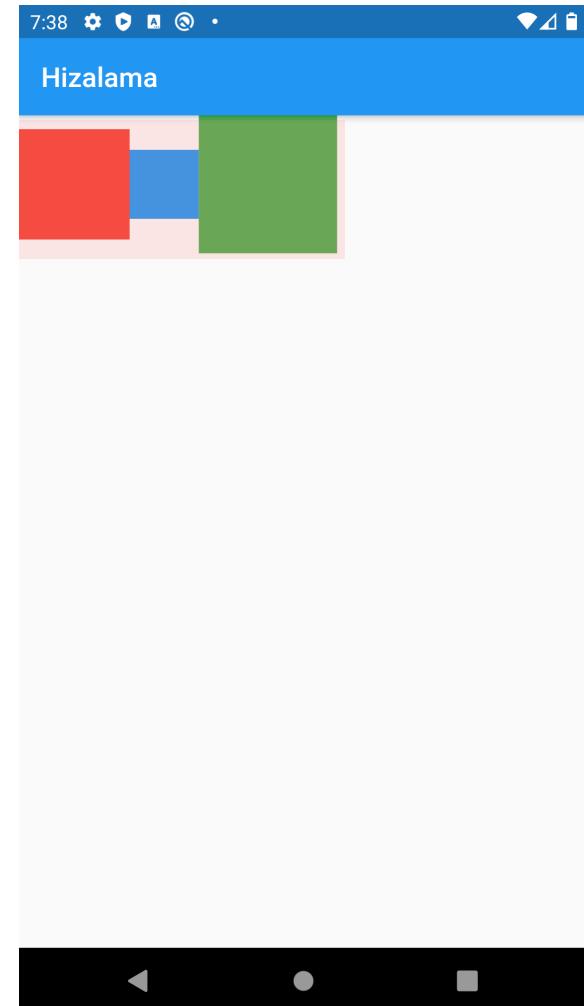


MainAxisSize

MainAxisSize

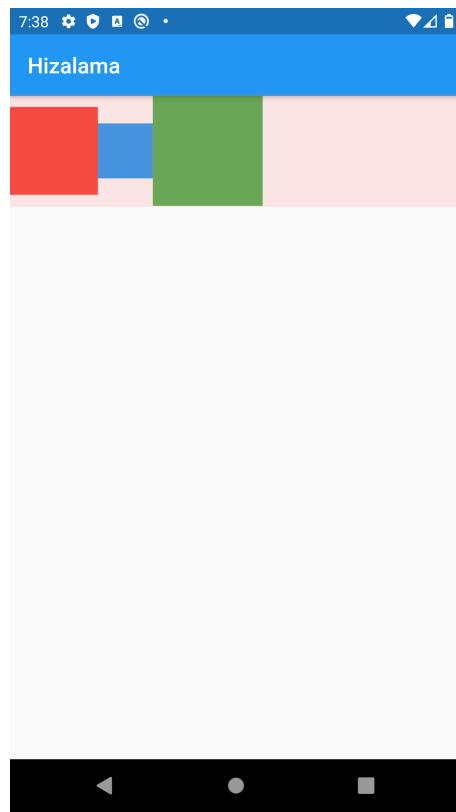
- Row veya Column içinde bulunduğu alanın boyutunu belirler.
- MainAxisSize.max aslında match parent
- MainAxisSize.min aslında wrap content

```
body: Row(  
    mainAxisAlignment: MainAxisAlignment.min,  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```

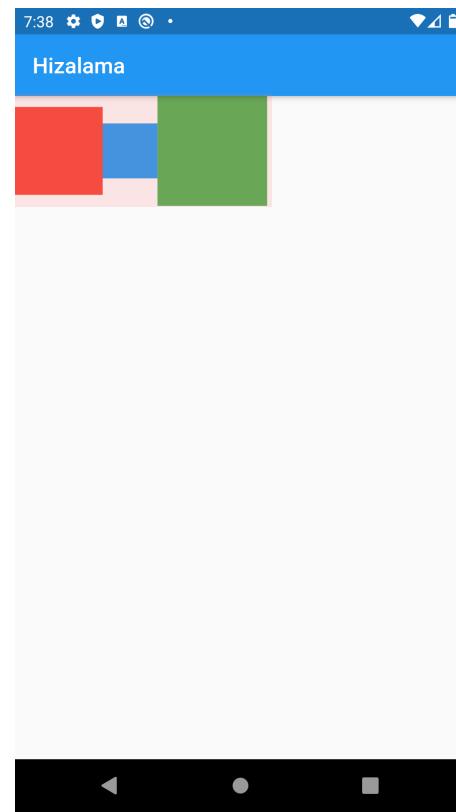


MainAxisSize - Row

max

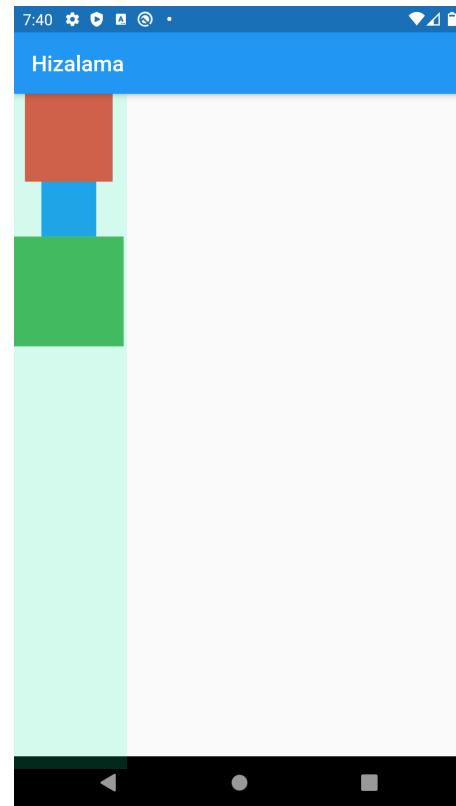


min

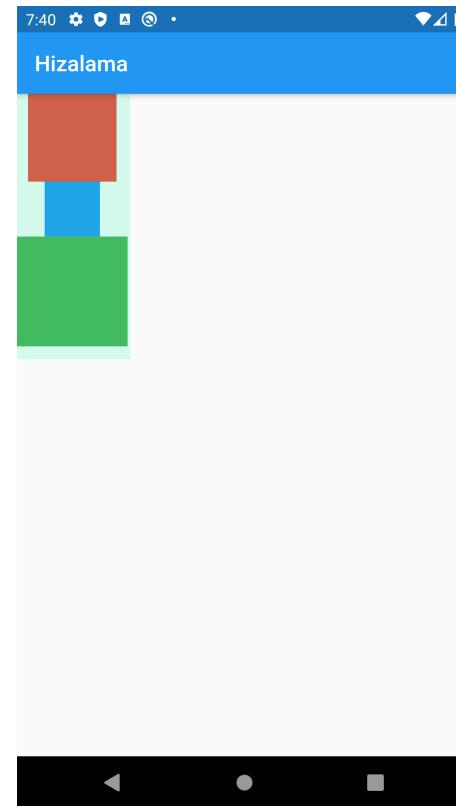


MainAxisSize - Column

max



min



Center

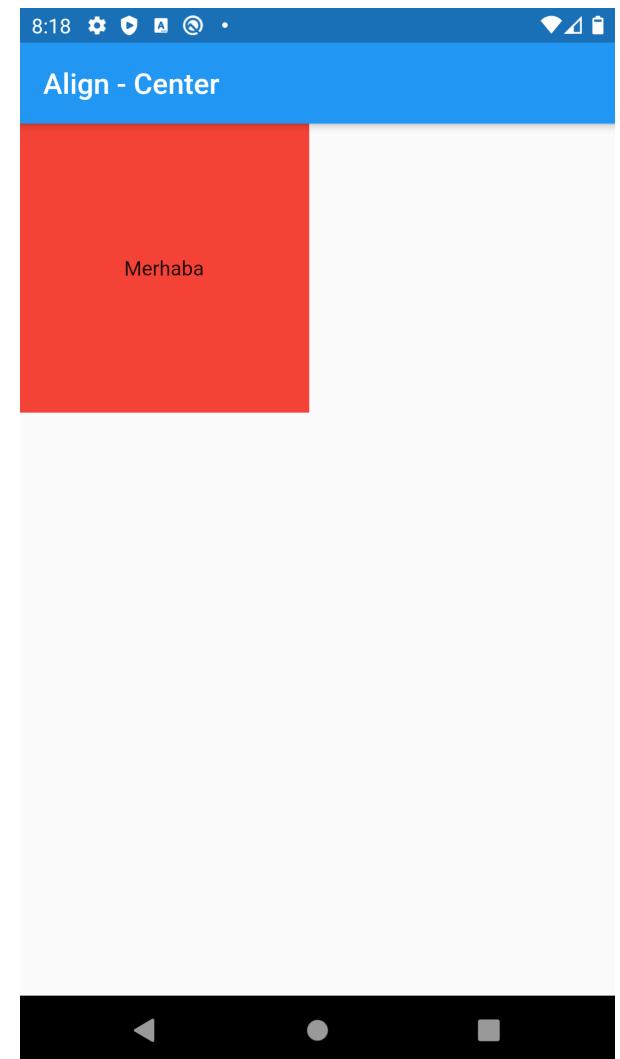
Center

- Widgetları içinde bulunduğu alana göre yatay ve dikeyde merkezler.

```
body: Row(  
  children: <Widget>[  
    Container(  
      width: 200,  
      height: 200,  
      color: Colors.red,  
      child: Center(  
        child: Text("Merhaba"),  
      ), // Center  
    ), // Container  
  ], // <Widget>[]  
, // Row
```

Text Container içinde yer alıyor.
Yer aldığı alana göre merkeze geçer.

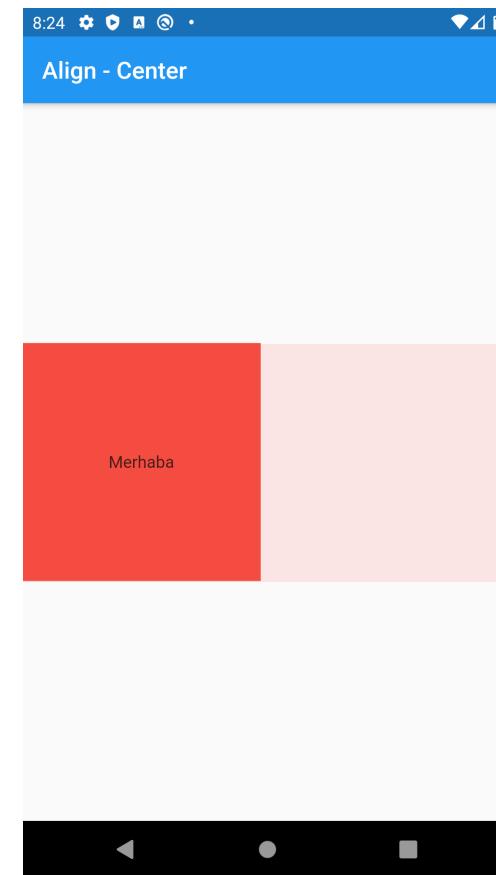
Kasım ADALAN



Center Örnek

```
body: Center(  
  child: Row(  
    children: <Widget>[  
      Container(  
        width: 200,  
        height: 200,  
        color: Colors.red,  
        child: Center(  
          child: Text("Merhaba"),  
        ), // Center  
      ), // Container  
    ], // <Widget>[]  
, // Row  
, // Center
```

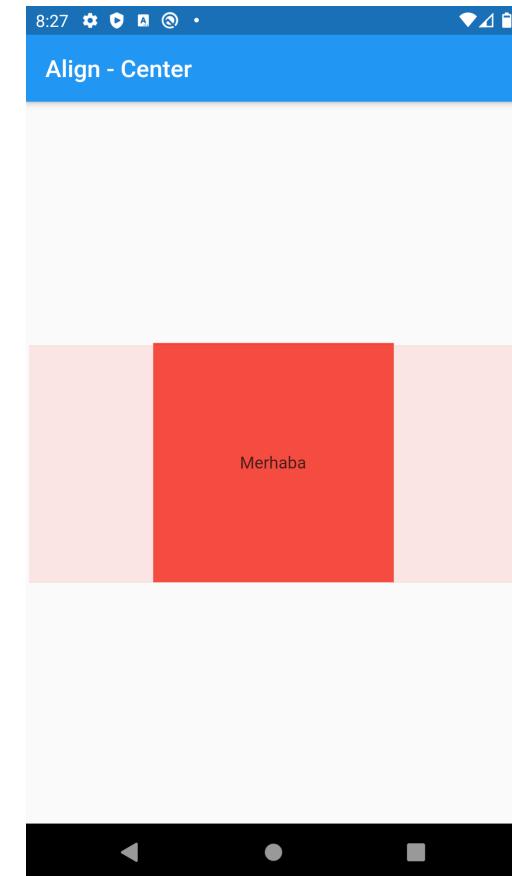
Row yatayda sayfayı kapladığı
için sadece dikeyde row
alanını ortalaşabilir.
Row içine ayrı müdahale
etmeliyiz.



Center Örnek

```
body: Center(  
    child: Row(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
            Container(  
                width: 200,  
                height: 200,  
                color: Colors.red,  
                child: Center(  
                    child: Text("Merhaba"),  
                ), // Center  
            ), // Container  
        ], // <Widget>[]  
    ), // Row  
, // Center
```

Row içinde ortalarsak yatay ve dikeyde ortada olur.

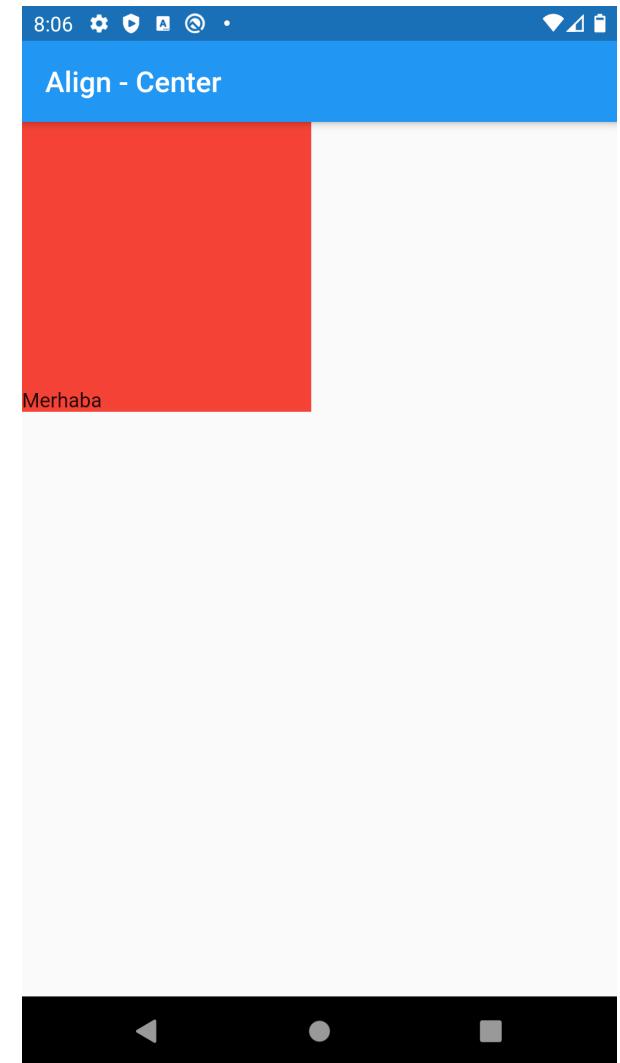


Align

Align

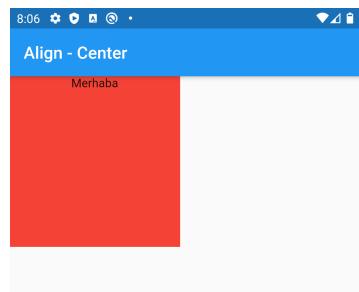
- İçinde bulunduğu alana göre hizalama yapar.
- Center yapısının daha detaylı hali olarak düşünülebilir.

```
body: Row(  
    children: <Widget>[  
        Container(  
            width: 200,  
            height: 200,  
            color: Colors.red,  
            child: Align(  
                alignment: Alignment.bottomLeft,  
                child: Text("Merhaba"),  
            ), // Align  
        ), // Container  
    ], // <Widget>[]  
, // Row
```

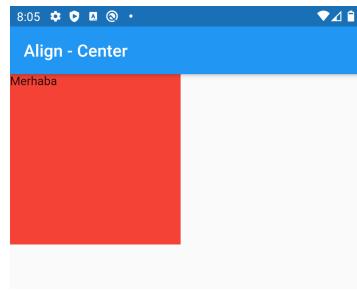


Align Örnekler

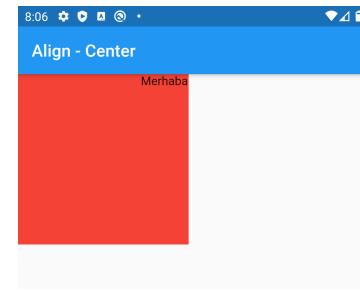
topcenter



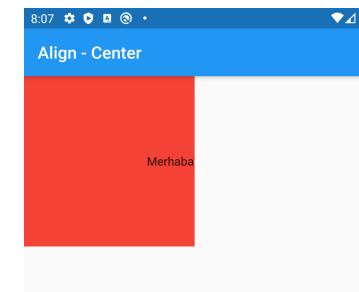
topleft



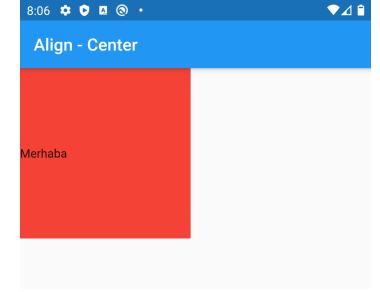
topright



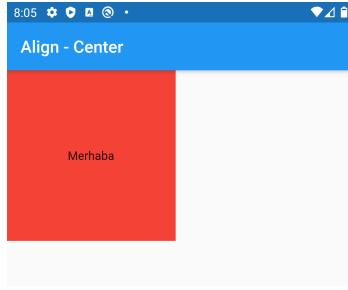
centerright



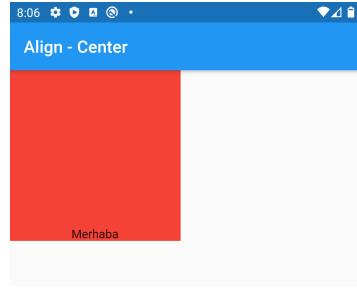
centerleft



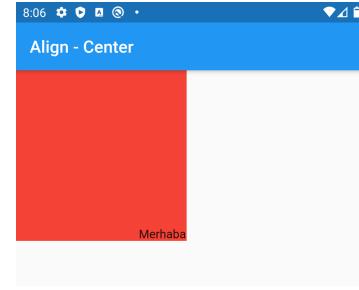
center



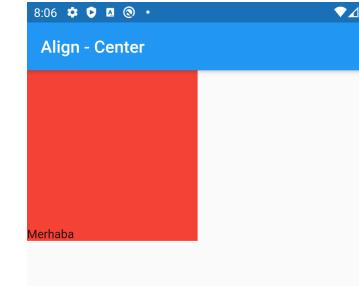
bottomcenter



bottomright



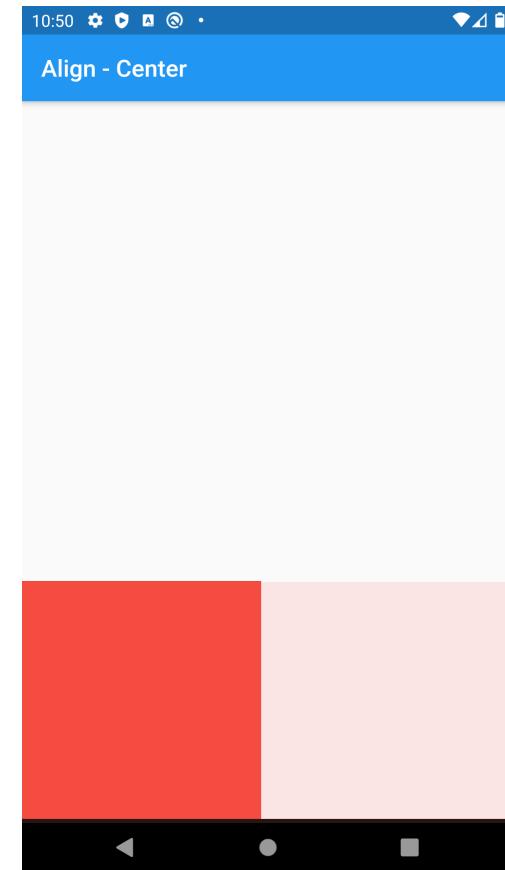
bottomleft



Align Örnekler

Row yatayda sayfayı kapladığı
için sadece dikeyde row
alanını ortalayabilir.
Row içine ayrı müdahale
etmeliyiz.

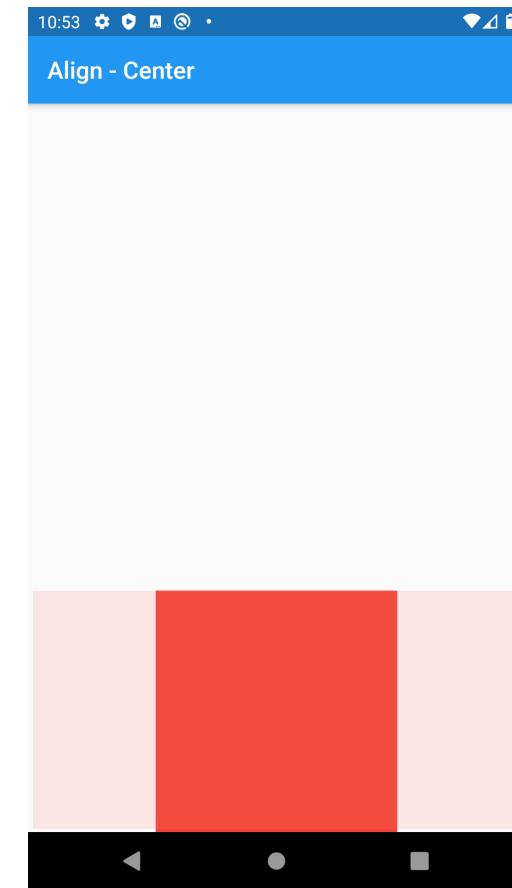
```
body: Align(  
  alignment: Alignment.bottomCenter,  
  child: Row(  
    children: <Widget>[  
      Container(width: 200, height: 200, color: Colors.red,  
        ), // Container  
      ], // <Widget>[]  
    ), // Row  
  ), // Align
```



Align Örnekler

```
body: Align(  
  alignment: Alignment.bottomCenter,  
  child: Row(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
      Container(width: 200, height: 200, color: Colors.red,  
        ), // Container  
    ], // <Widget>[]  
  ), // Row  
, // Align
```

Row içinde ortalarsak yatay
ve dikeyde ortada olur.

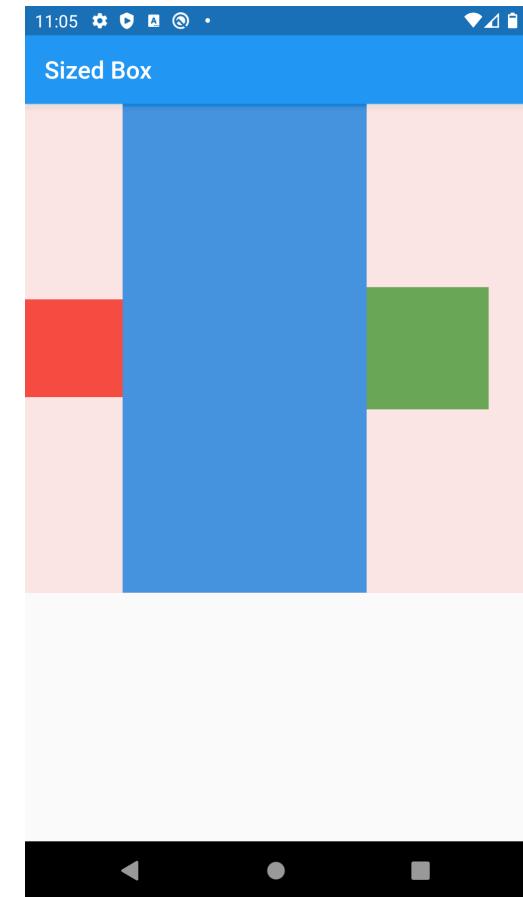


SizedBox

SizedBox : Tekrar Boyutlandırma

- Herhangi bir görsel nesneyi tekrar boyutlandıramızı.

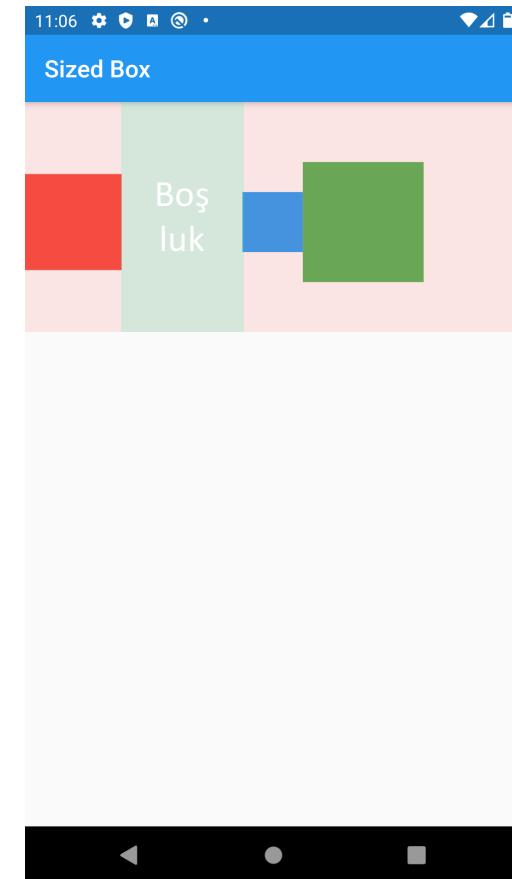
```
body: Row(  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        SizedBox(  
            width: 200,//Genişlik  
            height: 400,//Yükseklik  
            child: Container(width: 50,height: 50,color : Colors.blue),  
        ), // SizedBox  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```



SizedBox : Boşluk Oluşturma

- Diğer özelliği ise istenilen boyutta widgetlar arasında boşluk oluşturulabiliriz.

```
body: Row(  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        SizedBox(  
            width: 100,//Genişlik  
            height: 200,//Yükseklik  
        ), // SizedBox  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```

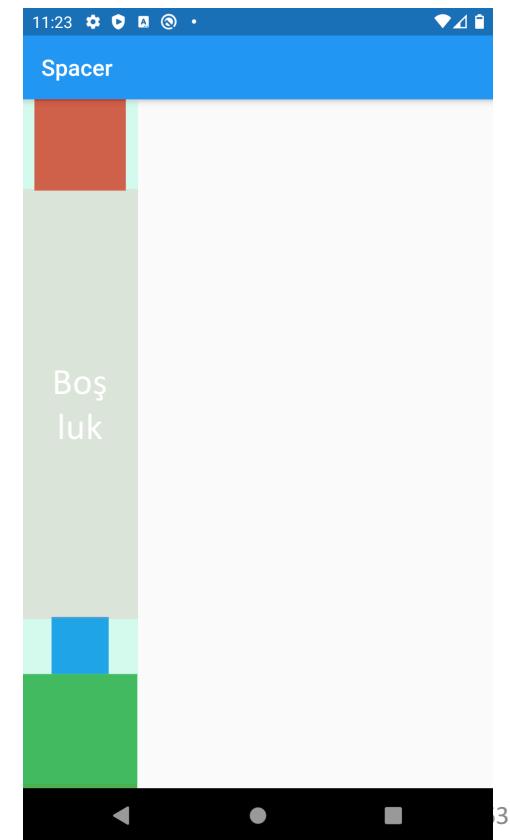


Spacer

Spacer : Boşluk

- Var olan boş alanı tam olarak kullanarak boşluk oluşturur.

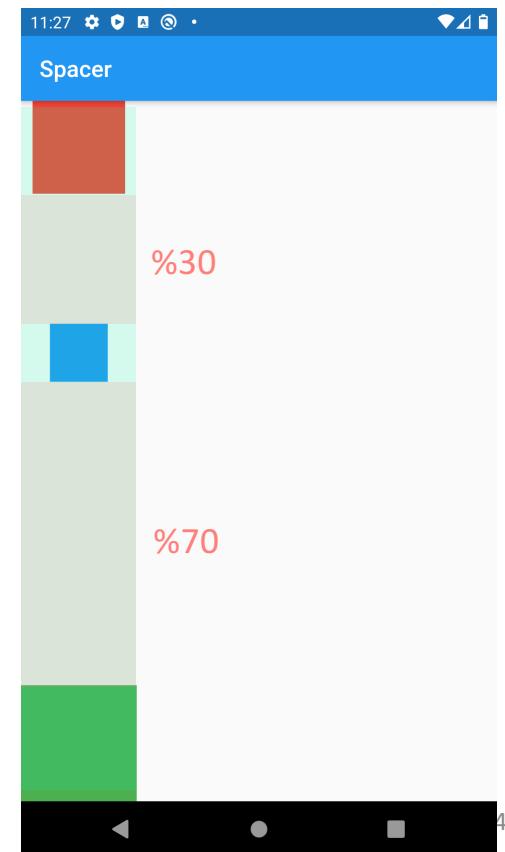
```
body: Column(  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        Spacer(),  
        Container(width: 50,height: 50,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Column
```



Spacer : Boşluk

- Oranlı boşluk verebiliriz bunun için flex özelliğini kullanabiliriz.

```
body: Column(  
    children: <Widget>[  
        Container(width: 80,height: 80,color : Colors.red),  
        Spacer(flex: 30,), %30  
        Container(width: 50,height: 50,color : Colors.blue),  
        Spacer(flex: 70,), %70  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
, // Column
```

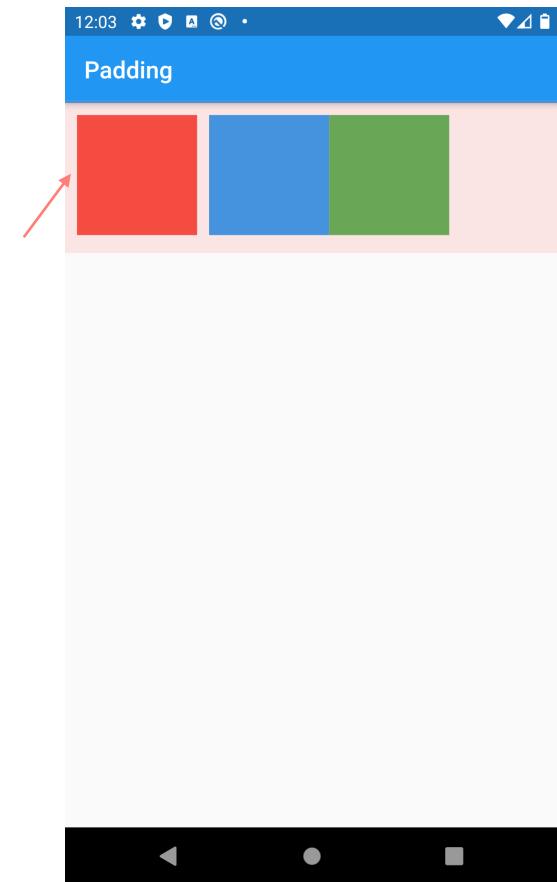


Padding

Padding

- Padding , widget etrafında boşluk oluşturmak için kullanılır.
- *EdgeInsets.all()* özelliği bütün kenarlara aynı oranda boşluk verir.

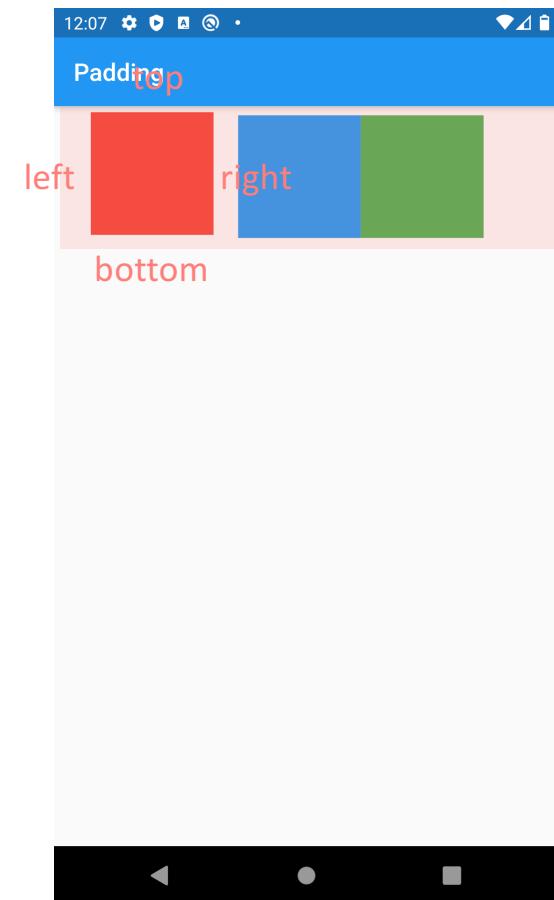
```
body: Row(  
    children: <Widget>[  
        Padding(  
            padding: const EdgeInsets.all(10.0),  
            child: Container(width: 100,height: 100,color : Colors.red),  
        ), // Padding  
        Container(width: 100,height: 100,color : Colors.blue),  
        Container(width: 100,height: 100,color : Colors.green),  
    ], // <Widget>[]  
>, // Row
```



Padding

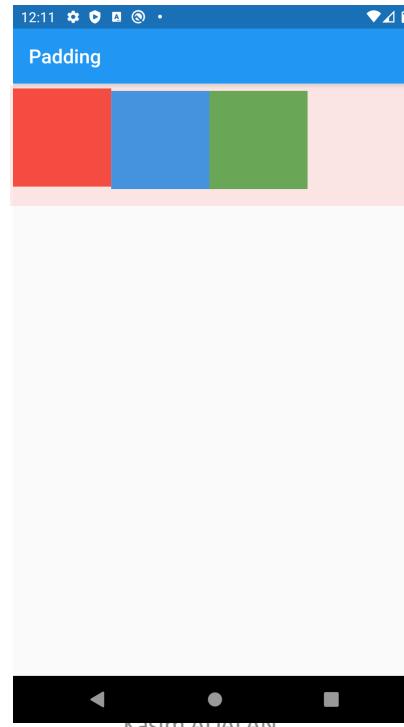
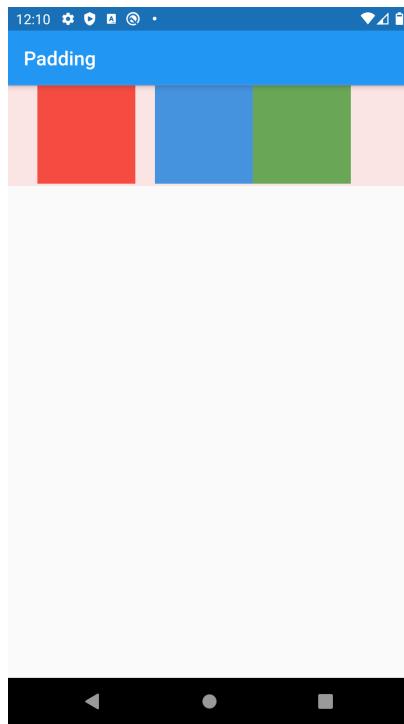
- *EdgeInsets.only()* , left,right,top,bottom şeklinde etrafına farklı oranlarda boşluk verebiliriz.

```
body: Row(  
  children: <Widget>[  
    Padding(  
      padding: const EdgeInsets.only(left: 30.0,right: 20.0,top:5.0,bottom:10.0),  
      child: Container(width: 100,height: 100,color : Colors.red),  
    ), // Padding  
    Container(width: 100,height: 100,color : Colors.blue),  
    Container(width: 100,height: 100,color : Colors.green),  
  ], // <Widget>[]  
, // Row
```

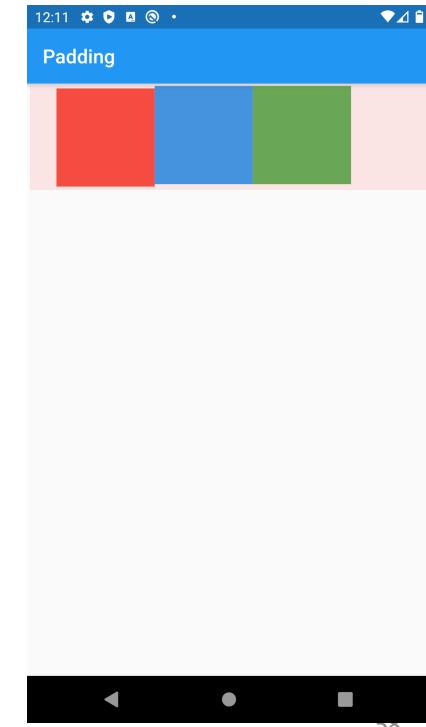


Padding

```
padding: const EdgeInsets.only(left: 30.0,right: 20.0), padding: const EdgeInsets.only(top:5.0,bottom:10.0),
```



```
padding: const EdgeInsets.only(left: 30.0,top:5.0),
```

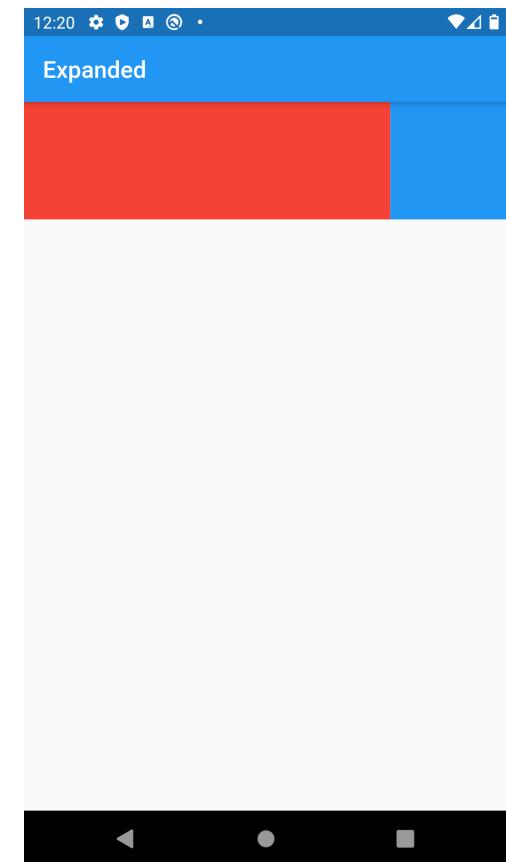


Expanded

Expanded

- Widget bulunduğu alanda Row içindeyse yatayda , Column içindeyse dikeyde boş olan alanda yayılır.

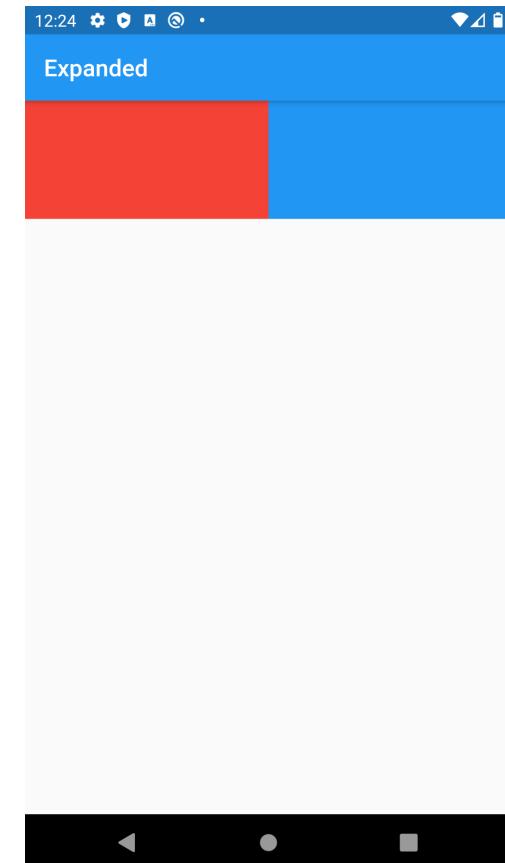
```
body: Row(  
    children: <Widget>[  
        Expanded(  
            child : Container(width: 100,height: 100,color : Colors.red),  
        ), // Expanded  
        Container(width: 100,height: 100,color : Colors.blue),  
    ], // <Widget>[]  
, // Row
```



Expanded

- Eğer birden fazla expanded olursa kendi aralarında eşit oranda yayılırlar.

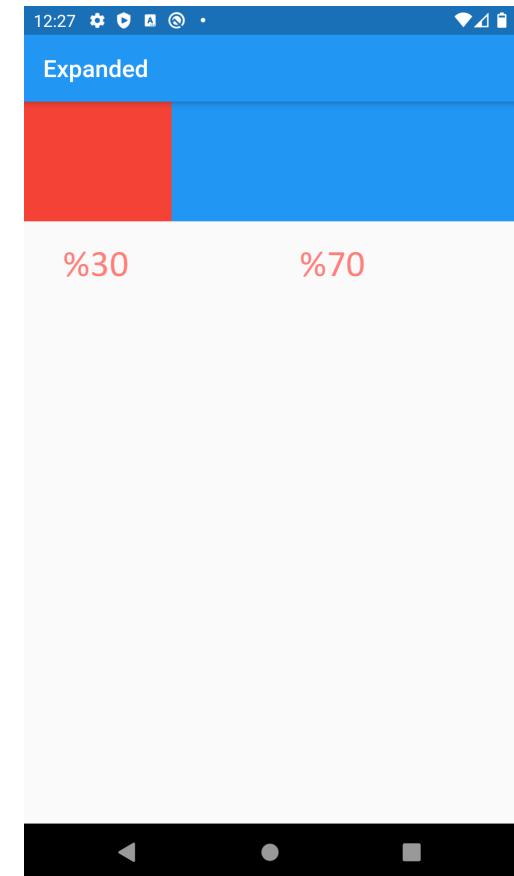
```
body: Row(  
    children: <Widget>[  
        Expanded(  
            child : Container(width: 100,height: 100,color : Colors.red),  
        ), // Expanded  
        Expanded(  
            child : Container(width: 100,height: 100,color : Colors.blue),  
        ), // Expanded  
    ], // <Widget>[]  
>, // Row
```



Expanded : flex (oranlama)

- Yayılma miktarı için oranlama yapılabilir.
- Oranlama için flex özelliği ile kullanılabilir.

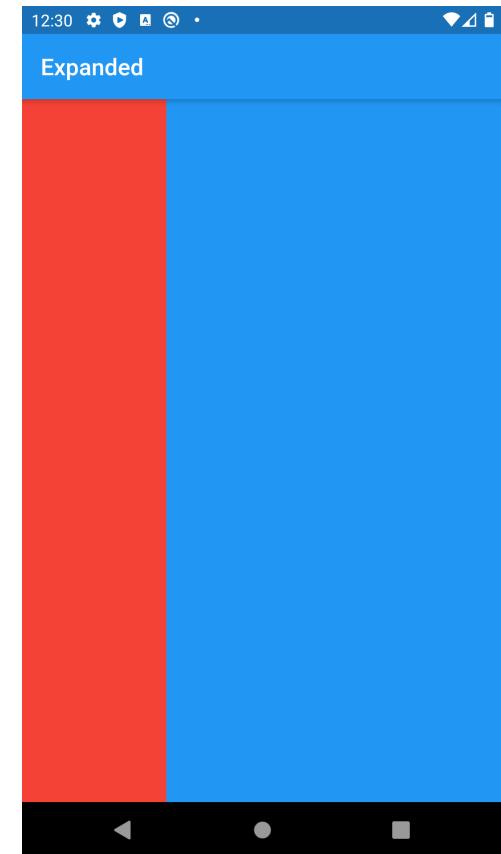
```
body: Row(  
  children: <Widget>[  
    Expanded(  
      flex: 30, %30  
      child : Container(width: 100,height: 100,color : Colors.red),  
    ), // Expanded  
    Expanded(  
      flex: 70, %70  
      child : Container(width: 100,height: 100,color : Colors.blue),  
    ), // Expanded  
  ], // <Widget>[]  
, // Row
```



Expanded

- Widget bulunduğu alanda Row içindeyse yatayda , Column içindeyse dikeyde boş olan alanda yayılır.
- Ters yönde yayılma yapmak için CrossAxisAlignmentAlignment.stretch özelliği kullanılır.

```
body: Row(  
    crossAxisAlignment: CrossAxisAlignment.Alignment.stretch,  
    children: <Widget>[  
        Expanded(  
            flex: 30,  
            child : Container(width: 100,height: 100,color : Colors.red),  
        ), // Expanded  
        Expanded(  
            flex: 70,  
            child : Container(width: 100,height: 100,color : Colors.blue),  
        ), // Expanded  
    ], // <Widget>[]  
>, // Row
```



Özelleştirilmiş Widget Oluşturma

Özelleştirilmiş Widget Oluşturma

- Sık kullandığınız tasarımları modüler olarak kullanmak için ayrı bir widget gibi oluşturup ismi ile erişim sağlayabiliriz.

```
class MaviKare extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
  
    return Container(  
      width: 50,  
      height: 50,  
      color: Colors.blue  
    );  
  }  
}
```

```
class Yazi extends StatelessWidget {  
  
  final String icerik;  
  final double yaziBoyut;  
  
  Yazi(this.icerik, this.yaziBoyut);  
  
  @override  
  Widget build(BuildContext context) {  
  
    return Text(icerik,  
      style: TextStyle(  
        fontSize: yaziBoyut,  
      ), // TextStyle  
    ); // Text  
  }  
}
```

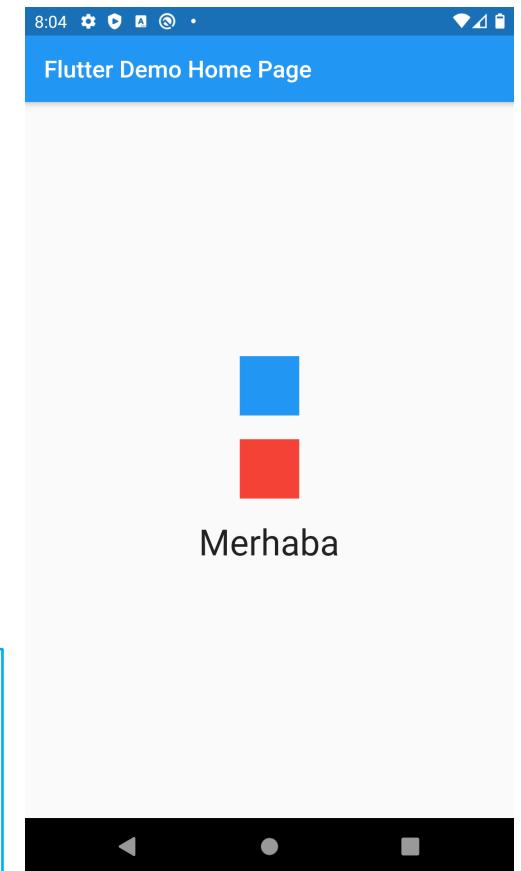
Özelleştirilmiş Widget Oluşturma

```
class MaviKare extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
  
    return Container(  
      width: 50,  
      height: 50,  
      color: Colors.blue  
    );  
  }  
}
```

```
class Yazi extends StatelessWidget {  
  
  String icerik;  
  double yaziBoyut;  
  
  Yazi(this.icerik, this.yaziBoyut);  
  
  @override  
  Widget build(BuildContext context) {  
  
    return Text(icerik,  
      style: TextStyle(  
        fontSize: yaziBoyut,  
      ), // TextStyle  
    ); // Text  
  }  
}
```

```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Center(  
        child: Column(  
          mainAxisAlignment: MainAxisAlignment.center,  
          children: <Widget>[  
            MaviKare(),  
            SizedBox(height: 20),  
            KirmiziKare(),  
            SizedBox(height: 20),  
            Yazi("Merhaba", 30.0),  
          ], // <Widget>[]  
        ), // Column  
      ), // Center  
    ); // Scaffold  
  }  
}
```

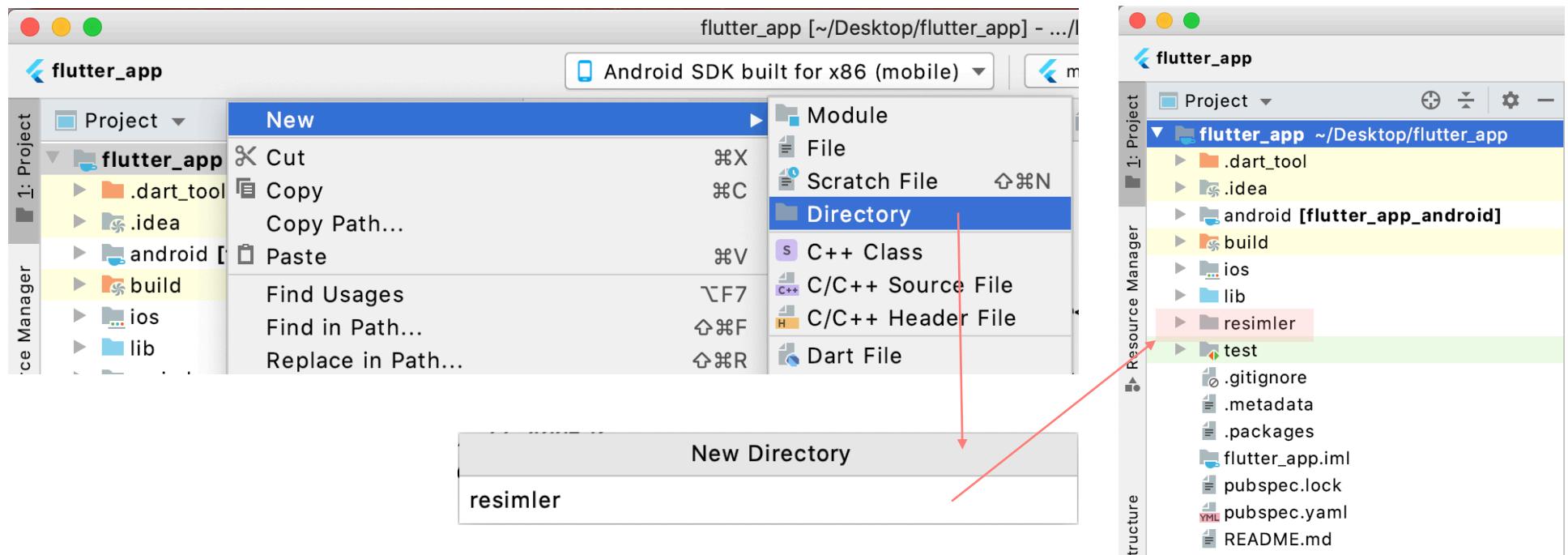
```
class KirmiziKare extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
  
    return Container(  
      width: 50,  
      height: 50,  
      color: Colors.red  
    );  
  }  
}
```



Projeye Resim Ekleme

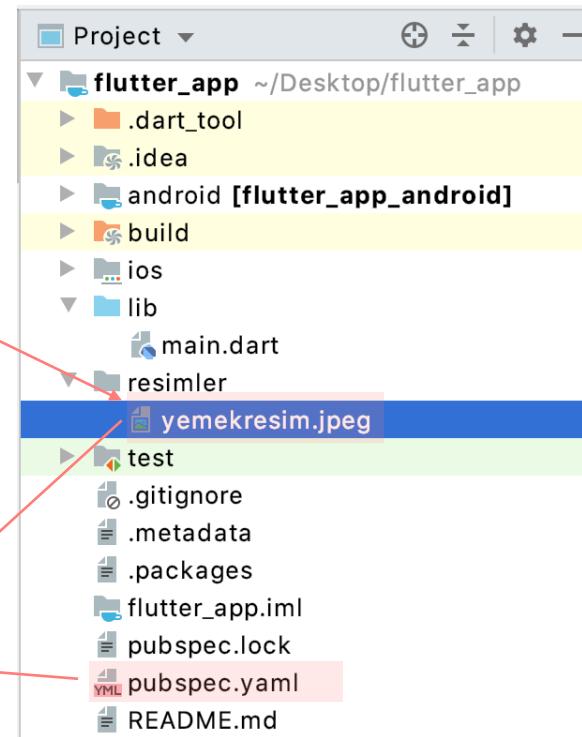
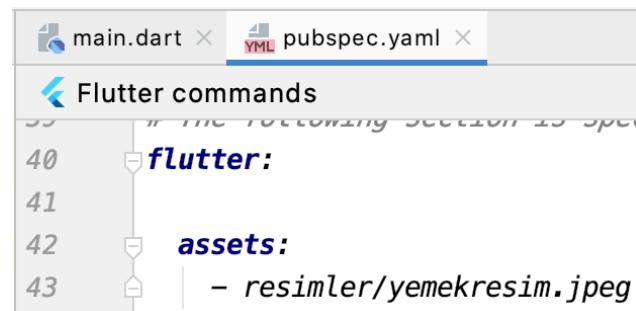
Resim Ekleme

- Resimler için dosya oluşturulur.

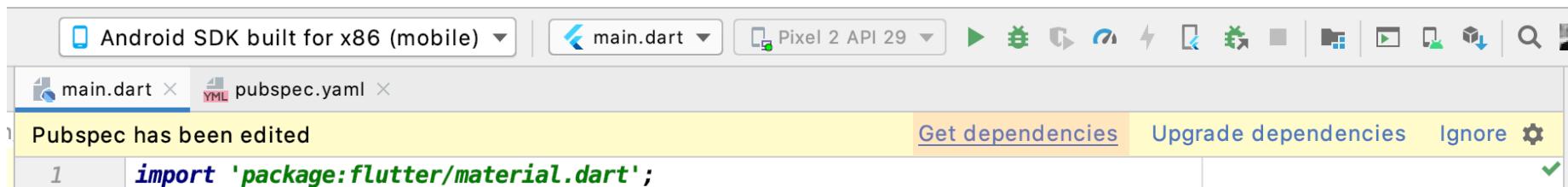


Resim Ekleme

- Dosya içine resim eklenir.
- Eklenen resim **pubspec.yaml** dosyasına eklenir.



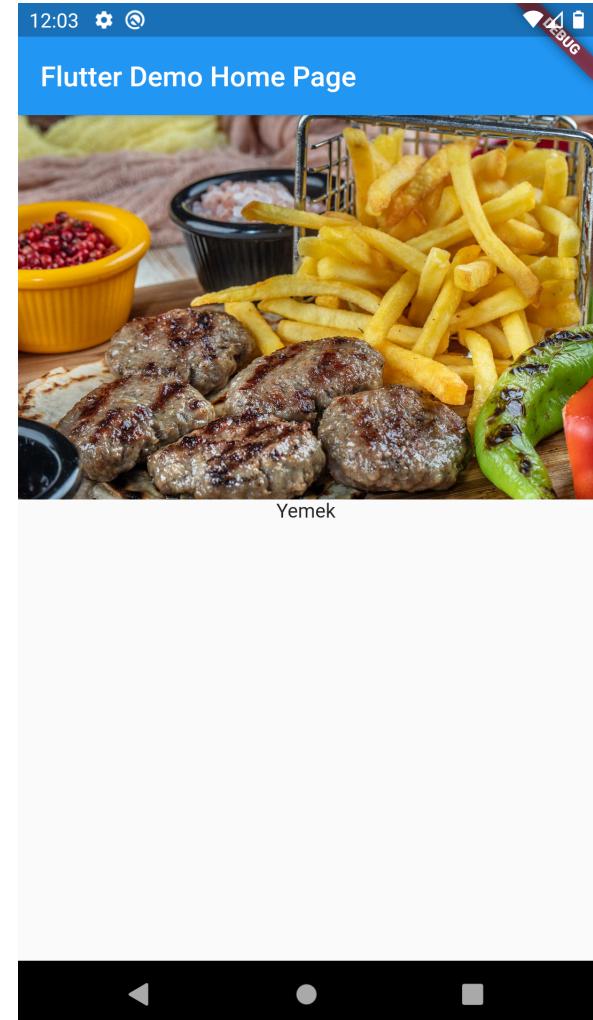
- Dart dosyası açılınca Sarı uyarı alanından Get dependencies seçilir.



Resim kullanımı

```
class _TasarimDeneme6 extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Column(  
        children: <Widget>[  
          Image.asset('resimler/yemekresim.jpeg'),  
          Text("Yemek"),  
        ], // <Widget>[]  
      ), // Column  
    ); // Scaffold  
  }  
}
```

Kasım ADALAN



Çoklu Ekran Desteği

Çoklu Ekran Desteği

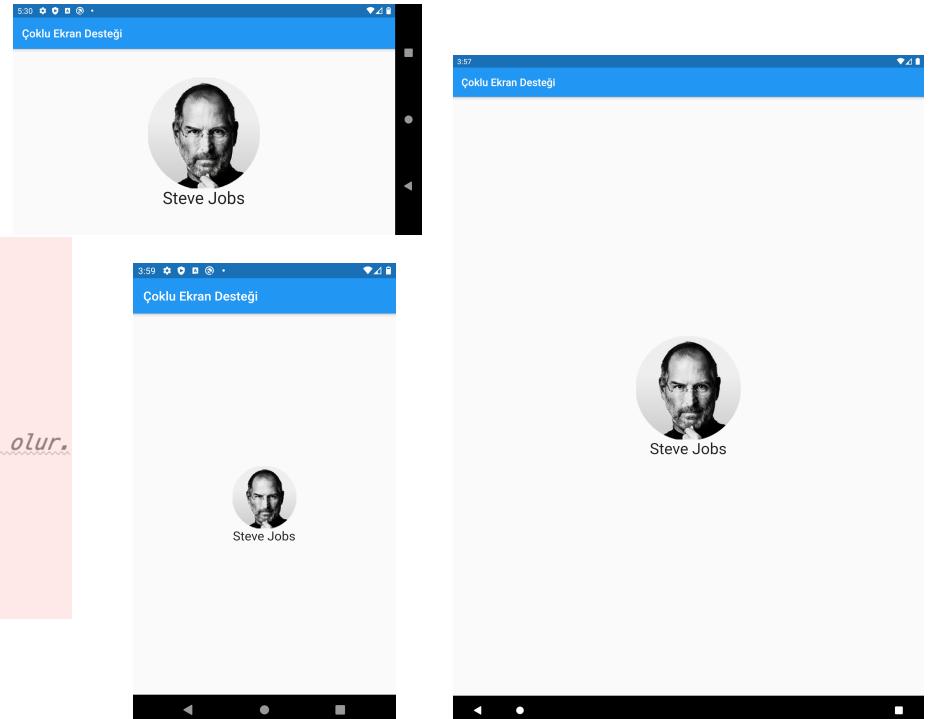
- 2 Yöntem vardır
- Her ekrana göre tasarım yapmak.
- Tasarımı ekrana göre oranlamak.

Her Ekrana Göre Tasarım Yapma

Her Ekrana Göre Tasarım Yapma

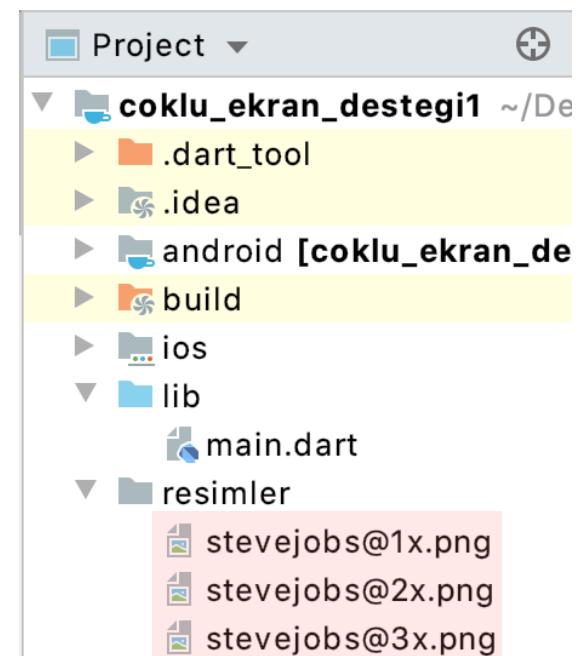
- Hangi ekran boyutunda olduğumuzu **LayoutBuilder** yapısı ile öğrenebiliriz.
- Boyuta göre tasarım oluşturarak kullanıcıya tasarımlı gösterebiliriz.

```
class _MyHomePageState extends State<MyHomePage> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: LayoutBuilder(  
        builder: (BuildContext context, BoxConstraints constraints) {  
          if(constraints.maxWidth < 600){//Ekran genişliği 600 den küçükse  
            //Telefon tasarımı  
  
            }  
          else{  
            //600 dan büyük olursa tablet veya telefonu yan yatırma (landscape) işlemi olur.  
            //Tablet tasarımı  
          }  
        }, // LayoutBuilder  
    ); // Scaffold  
  }  
}
```



Her Ekrana Göre Tasarım Yapma

- Eğer resim kullanıyorsak aynı resmi ekran boyutlarına göre farklı boyutlarda oluşturmamız tasarım için faydalı olmaktadır.
- Büyük çözünürlükteki resmi daha büyük ekranlarda , küçük çözünürlükteki resmi daha küçük ekranlarda gösterebiliriz.
- Resimlere isim verilirken çözünürlük ile ilgili bir isimlendirme iyi olacaktır.



Her Ekrana Göre Tasarım Yapma

- Her ekrana göre tasarım oluşturma.

```
class TabletTasarim extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
          Image.asset("resimler/stevejobs@2x.png"),  
          Text("Steve Jobs",  
            style: TextStyle(  
              fontSize: 30.0,  
            ), // TextStyle  
            ), // Text  
          ], // <Widget>[]  
        ), // Column  
      ); // Center  
  }  
}
```

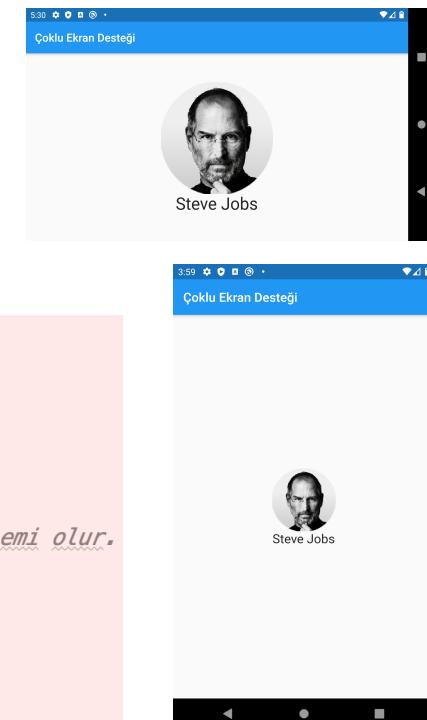
```
class TelefonTasarim extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: Column(  
        mainAxisAlignment: MainAxisAlignment.center,  
        children: <Widget>[  
          Image.asset("resimler/stevejobs@1x.png"),  
          Text("Steve Jobs",  
            style: TextStyle(  
              fontSize: 20.0,  
            ), // TextStyle  
            ), // Text  
          ], // <Widget>[]  
        ), // Column  
      ); // Center  
  }  
  } ADALAN }
```

Her Ekrana Göre Tasarım Yapma

- Tasarımların ekrana göre açılmasını sağlamalıyız.
- Tasarım geçisi otomatik gerçekleşmektedir.

```
class _MyHomePageState extends State<MyHomePage> {

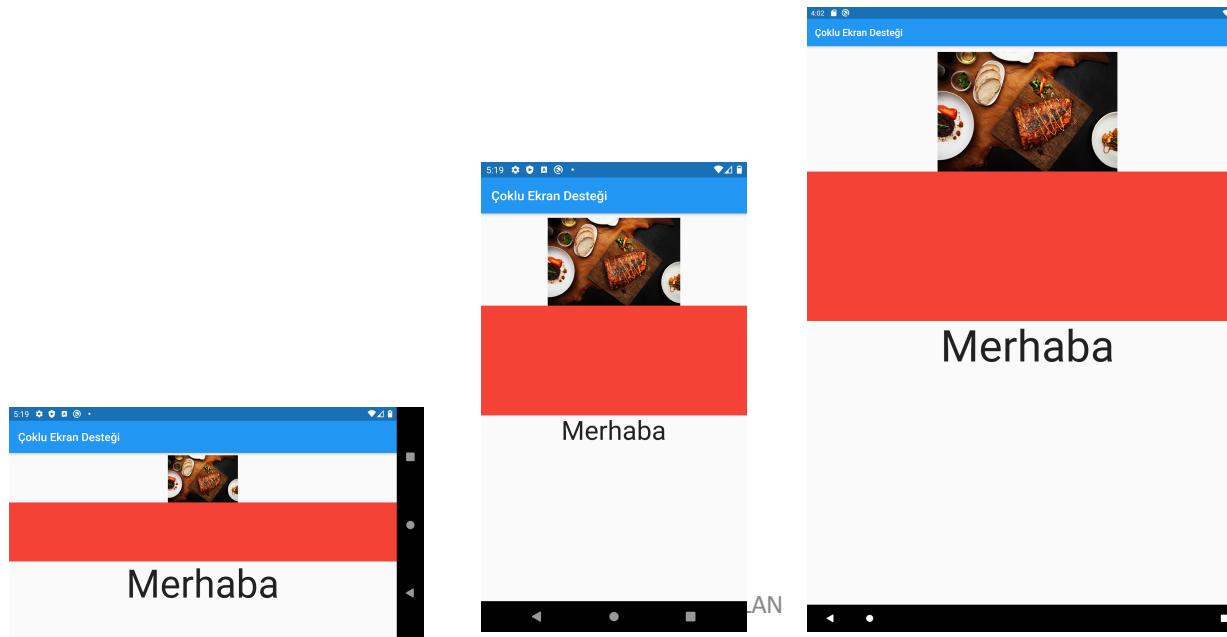
    @override
    Widget build(BuildContext context) {
        return Scaffold(
            appBar: AppBar(
                title: Text(widget.title),
            ), // AppBar
            body: LayoutBuilder(
                builder: (BuildContext context, BoxConstraints constraints) {
                    if(constraints.maxWidth < 600){//Ekran genişliği 600 den küçükse
                        //Telefon tasarım
                        return TelefonTasarim();
                    }else{
                        //600 dan büyük olursa tablet veya telefonu yan yatırma (landscape) işlemi olur.
                        //Tablet tasarım
                        return TabletTasarim();
                    }
                },
            ), // LayoutBuilder
        ); // Scaffold
    }
}
```



Tasarımı Ekrana Göre Oranlama

Tasarımı Ekrana Göre Oranlama

- Tasarımda sabit olarak oluşturduğumuz genişlik , yükseklik , boşluk miktarı gibi sayıları ekranın boyutuna oranlayarak işlem yapabiliriz.
- Bu şekilde yapılan oranlama dinamik olarak boyutlama sağlamaktadır.
- Örneğin : Butonun genişliğini ekranın yarısı kadar olsun diyebiliriz.



Tasarımı Ekrana Göre Oranlama

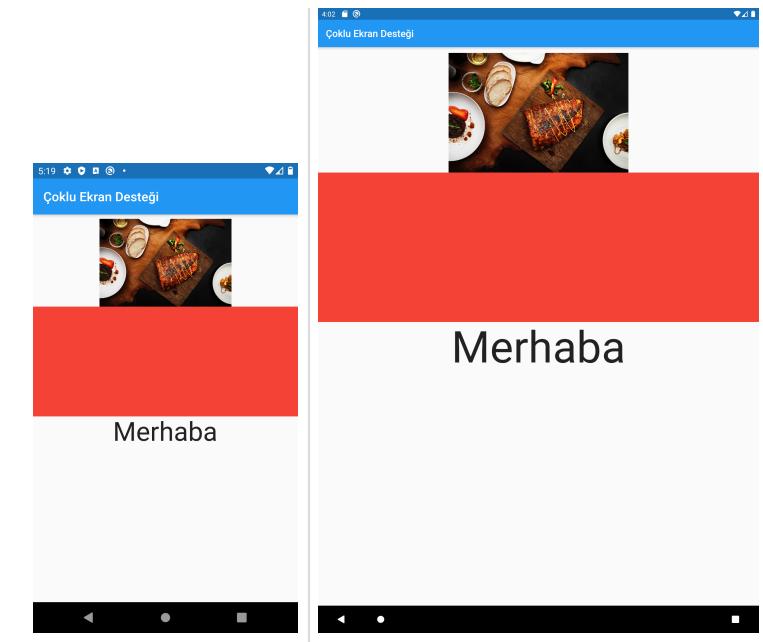
- Oranlama işlemi için ekran bilgisini **MediaQuery** sınıfı ile alabiliriz.

```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {

    var ekranBilgisi = MediaQuery.of(context);
    final double ekranYuksekligi = ekranBilgisi.size.height;
    final double ekranGenisligi = ekranBilgisi.size.width;

    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ), // AppBar
      body: Column(
        children: <Widget>[
          Padding(
            //const özelliğini silmemiz gereklidir çünkü final bir değişken ekliyoruz.
            padding: EdgeInsets.only(top:ekranYuksekligi/100),

```



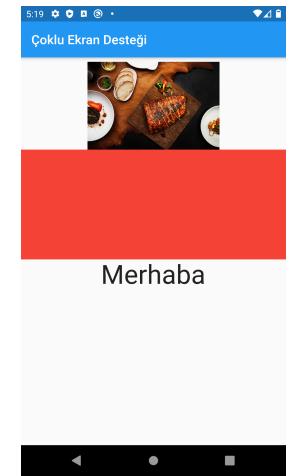
Tasarımı Ekrana Göre Oranlama

```
class _MyHomePageState extends State<MyHomePage> {  
  @override  
  Widget build(BuildContext context) {  
    // Ekran bilgileri alınır  
  
    var ekranBilgisi = MediaQuery.of(context);  
    final double ekranYuksekligi = ekranBilgisi.size.height;  
    final double ekranGenisligi = ekranBilgisi.size.width;  
  
    return Scaffold(  
      appBar: AppBar(  
        title: Text(widget.title),  
      ), // AppBar  
      body: Column(  
        children: <Widget>[  
          Padding(  
            //const özelliğini silmemiz gereklidir çünkü final bir değişken ekliyoruz.  
            padding: EdgeInsets.only(top:ekranYuksekligi/100), // Boşluk oranlama  
            child: SizedBox(  
              width: ekranGenisligi/2, // Boyut oranlama  
              height: ekranYuksekligi/5,  
              child: Image.asset("resimler/resim.jpg")  
            ), // SizedBox  
          ), // Padding  
        ],  
      ),  
    );  
  }  
}
```

```
  }, // Container  
  Container(  
    width: ekranGenisligi,  
    height: ekranYuksekligi/4,  
    color: Colors.red,  
  ), // Container  
  Text("Merhaba",  
    style: TextStyle(  
      fontSize: ekranGenisligi/10, // Font boyutu oranlama  
    ), // TextStyle  
  ), // Text  
], // <Widget>[]  
, // Column  
); // Scaffold  
}  
}
```

Boyut oranlama

Font boyutu oranlama



Merhaba

81

Tasarım Uygulamaları

UYGULAMA – İLHAM VER



```

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    var ekranBilgisi = MediaQuery.of(context);
    final double ekranYuksekligi = ekranBilgisi.size.height;
    final double ekranGenisligi = ekranBilgisi.size.width;

    return Scaffold(
      appBar: AppBar(
        title: Text("İlham Ver"),
      ), // AppBar
      body: Column(
        children: [
          Padding(
            padding: EdgeInsets.only(top: ekranYuksekligi/100, bottom: ekranYuksekligi/100),
            child: SizedBox(
              width: ekranGenisligi/4,
              child: Image.asset("resimler/stevejobs.png")
            ), // SizedBox
          ), // Padding
          Text("Steve Jobs",
            style: TextStyle(
              color: Colors.redAccent,
              fontWeight: FontWeight.bold,
              fontSize: ekranGenisligi/25,
            ), // TextStyle
          ), // Text
          Spacer(),
        ],
      ),
    );
  }
}

```



```

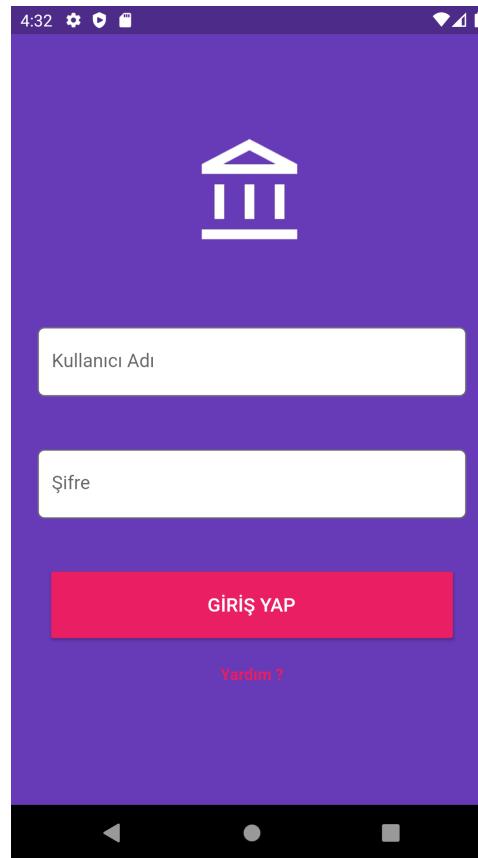
Padding(
  padding: EdgeInsets.only(left: ekranGenisligi/100,right: ekranGenisligi/100),
  child: Text("Dünyayı değiştirecek insanlar, onu değiştirebileceklerini düşünecek kadar çılgın olanlardır.",
    textAlign: TextAlign.center,
    style: TextStyle(
      fontSize: ekranGenisligi/25,
    ), // TextStyle
  ), // Text
), // Padding
Spacer(),
Padding(
  padding: EdgeInsets.only(bottom: ekranYuksekligi/100),
  child: SizedBox(
    width: ekranGenisligi/2,
    height: ekranYuksekligi/15,
    child: ElevatedButton(
      child: Text("İLHAM VER",style: TextStyle(fontSize: ekranGenisligi/25,color:Colors.white,)),
      onPressed: (){
        print("İlhgam verildi");
      },
      style: ElevatedButton.styleFrom(
        primary: Colors.redAccent,
      ),
    ), // ElevatedButton
  ), // SizedBox
), // Padding

```

Dünyayı değiştirecek insanlar, onu değiştirebileceklerini düşünecek kadar çılgın olanlardır.



UYGULAMA – LOG IN

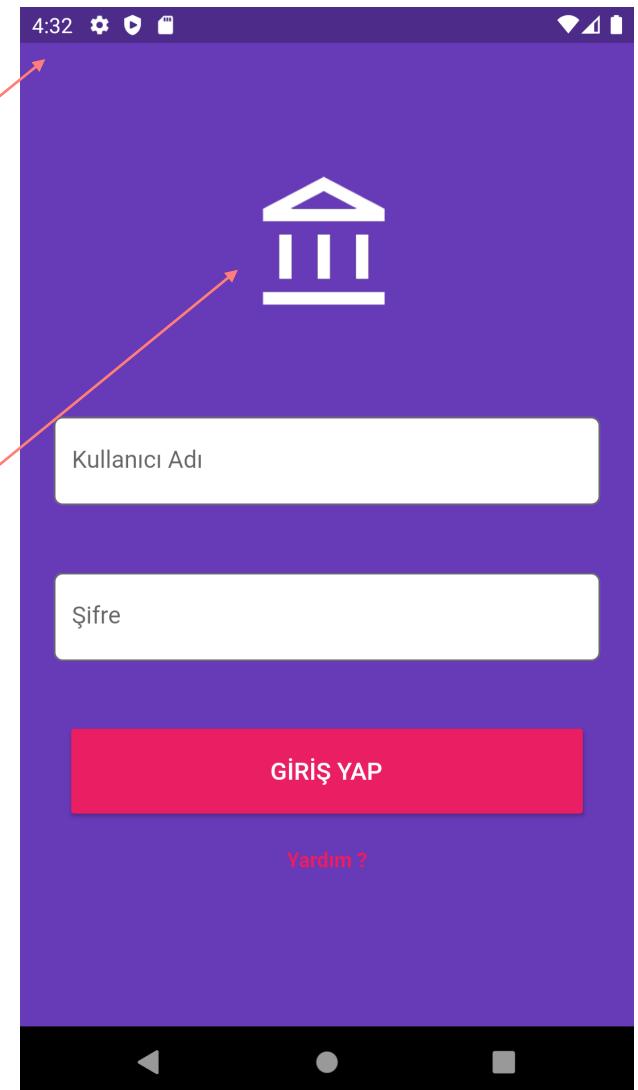


```
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {

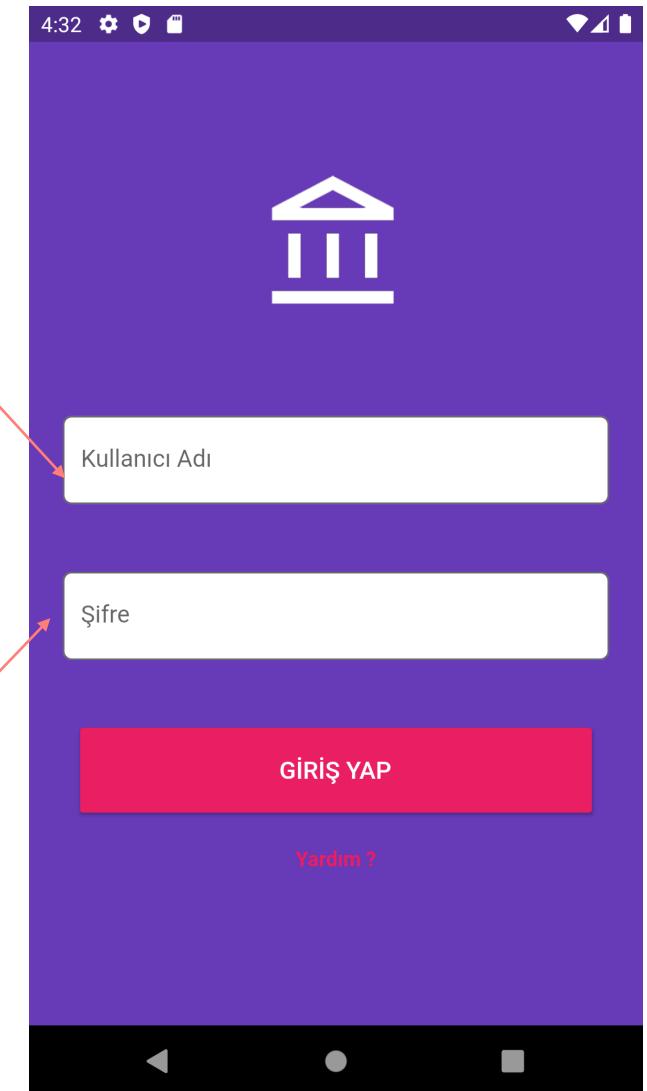
    var ekranBilgisi = MediaQuery.of(context);
    final double ekranYuksekligi = ekranBilgisi.size.height;
    final double ekranGenisligi = ekranBilgisi.size.width;

    return Scaffold(
      backgroundColor: Colors.deepPurple,
      //Tam ekran renklendirme için kullanıldı.
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Padding(
              padding: EdgeInsets.only(bottom : ekranYuksekligi/20),
              child: SizedBox(
                width: ekranGenisligi/4,
                height: ekranYuksekligi/6,
                child: Image.asset("resimler/logo.png")), // SizedBox
          ), // Padding
        ],
      ),
    );
  }
}
```

Kasım ADALAN

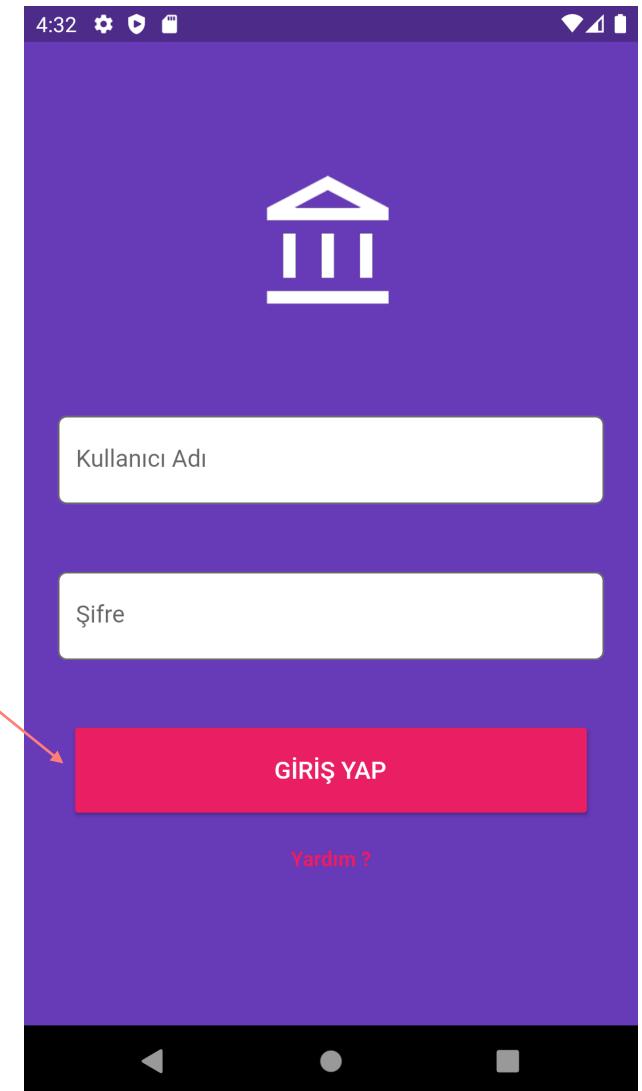


```
Padding(  
  padding: EdgeInsets.all(ekranYuksekligi/30),  
  child: TextField(  
    decoration: InputDecoration(  
      hintText: 'Kullanıcı Adı',  
      filled: true, //Arkaplan rengini aktif eder  
      fillColor: Colors.white,  
      border: OutlineInputBorder( //Çerçeve ekleme  
        borderRadius: BorderRadius.all(Radius.circular(ekranGenisligi/70)),  
      ), // OutlineInputBorder  
    ), // InputDecoration  
  ), // TextField  
, // Padding  
  
Padding(  
  padding: EdgeInsets.all(ekranYuksekligi/30),  
  child: TextField(  
    obscureText: true, //Şifre girerken saklama özelliği  
    decoration: InputDecoration(  
      hintText: 'Şifre',  
      filled: true, //Arkaplan rengini aktif eder  
      fillColor: Colors.white,  
      border: OutlineInputBorder(  
        borderRadius: BorderRadius.all(Radius.circular(ekranGenisligi/70)),  
      ), // OutlineInputBorder  
    ), // InputDecoration  
  ), // TextField  
, // Padding
```



```
Padding(  
  padding: EdgeInsets.all(ekranYuksekligi/30),  
  child: SizedBox(  
    width: ekranGenisligi/1.2,  
    height: ekranYuksekligi/12,  
    child: ElevatedButton(  
      child: Text("GİRİŞ YAP", style: TextStyle(fontSize: ekranGenisligi/25,color: Colors.white,),),  
      style:ElevatedButton.styleFrom(  
        onPrimary: Colors.pink,  
      ),  
      onPressed: (){  
        print("Giriş yapıldı");  
      }  
    ), // ElevatedButton  
  ), // SizedBox  
, // Padding  
  
GestureDetector(  
  onTap: (){  
    print("Yardım seçildi");  
  },  
  child: Text("Yardım?",  
    textAlign: TextAlign.center,  
    style: TextStyle(  
      color: Colors.pink,  
      fontWeight: FontWeight.bold,  
      fontSize: ekranGenisligi/30,  
    ), // TextStyle  
  ), // Text  
, // GestureDetector
```

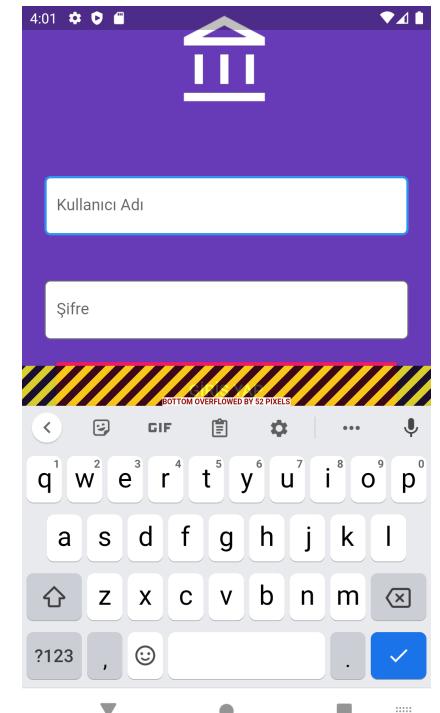
Kasım ADALAN



Overflow By Pixel Uyarısı Çözümü

- Bu uyarı ekran küçüldüğünde görsel nesnelerin görünmemesinden kaynaklı oluşmaktadır.
- Bunu çözmek için tasarıma scroll etme özelliği ekliyoruz.
- Ekran küçüldüğünde küçülen alanda yukarı aşağı scroll edebiliyoruz.

```
return Scaffold(  
    backgroundColor: Colors.deepPurple,  
    //Tam ekran renklendirme için kullanıldı.  
    body: Center(  
        child: SingleChildScrollView(  
            //Textfieldlara veri girme işlemi olurken ekran daralır  
            //ve bazı görseller görünmez bu olduğunda overflow by pixel uyarısı alırız.  
            //Bu widget sayesinde ekran küçülürse scroll özelliği aktif olur.  
            child: Column(  
                mainAxisAlignment: MainAxisAlignment.center,  
                children: <Widget>[  
                    Padding(  
                        padding: EdgeInsets.only(bottom : ekranYuksekligi/20),  
                        child: SizedBox(  
                            width: ekranGenisligi/4,  
                            height: ekranYuksekligi/6,  
                            child: Image.asset("resimler/logo.png")), // SizedBox  
                ), // Padding
```



UYGULAMA – YEMEK TARİFİ



Özel Text Oluşturma

```
//Özel bir text oluşturduk
class Yazi extends StatelessWidget {

    String icerik;
    double yaziBoyut;

    Yazi(this.icerik, this.yaziBoyut);

    @override
    Widget build(BuildContext context) {

        return Text(icerik,
            style: TextStyle(
                fontSize: yaziBoyut,
            ), // TextStyle
        ); // Text
    }
}
```

```

class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {

    var ekranBilgisi = MediaQuery.of(context);
    final double ekranYuksekligi = ekranBilgisi.size.height;
    final double ekranGenisligi = ekranBilgisi.size.width;

    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ), // AppBar
      body: SingleChildScrollView(
        child: Column(
          children: <Widget>[
            SizedBox(
              width: ekranGenisligi, //Yükseklikte genişlikle oranlı olarak artacak.
              child: Image.asset("resimler/yemekresim.jpeg")
            ), // SizedBox

```



```

Row(
  children: [
    Expanded(
      child: SizedBox(
        height: ekranGenisligi/8,
        child: TextButton(
          child: Yazi("Beğen", ekranGenisligi/25),
          style:TextButton.styleFrom(backgroundColor: Colors.orange,),
          onPressed: (){
            print("Beğenildi");
          },
        ), // TextButton
      ), // SizedBox
    ), // Expanded
    Expanded(
      child: SizedBox(
        height: ekranGenisligi/8,
        child: TextButton(
          child: Yazi("Yorum Yap", ekranGenisligi/25),
          style:TextButton.styleFrom(backgroundColor: Colors.deepOrangeAccent,),
          onPressed: (){
            print("Yorum yapıldı");
          },
        ), // TextButton
      ), // SizedBox
    ), // Expanded
  ],
), // Row

```



```

- Padding(
  padding: EdgeInsets.all(ekranYuksekligi/100),
  child: Column("//Ana Columndan farklı olarak bögesel olarak
// içerikleri sola hizalamak için column kullandık.
  mainAxisAlignment: MainAxisAlignment.start,
  children: <Widget>[
    Text("Köfte",
      style: TextStyle(
        color: Colors.deepOrangeAccent,
        fontWeight: FontWeight.bold,
        fontSize: ekranGenisligi/20,
      ), // TextStyle
    ), // Text
    Row(
      children: <Widget>[
        Yazi("Izgara Üzerinde Pişirime Uygun", ekranGenisligi/25),
        Spacer(),
        Yazi("8 Ağustos", ekranGenisligi/25),//Özel text
      ], // <Widget>[]
    ), // Row
  ], // <Widget>[]
), // Column
), // Padding

```



```
Padding(  
    padding: EdgeInsets.all(ekranYuksekligi/100),  
    child: Yazi("Köfte harçını hazırlamak için, soğanları rendeleyin ve maydanozları ince ince kiyın. " "  
    "İsterseniz, bir dış sarımsak da ekleyebilirsiniz."  
    "Soğan, maydanoz, kıyma, yumurta, zeytinyağı ve tuzu bir kaba alıp iyice yoğurun. "  
    "Bu sırada istedığınız baharatları da ekleyerek yoğurmaya devam edin."  
    "Hazırladığınız harçın üzerine streç filmle kapatarak yarı saat buz dolabında dinlendirin."  
    "Ardından harçtan ceviz büyülüğünde parçalar koparıp yuvarlayın. "  
    "1 cm olacak şekilde üzerine bastırarak yassılaştırın.",ekranGenisligi/25), // Yazi  
, // Padding
```

Köfte harçını hazırlamak için, soğanları rendeleyin ve maydanozları ince ince kiyın. İsterseniz, bir dış sarımsak da ekleyebilirsiniz.

Soğan, maydanoz, kıyma, yumurta, zeytinyağı ve tuzu bir kaba alıp iyice yoğurun. Bu sırada istedığınız baharatları da ekleyerek yoğurmaya devam edin.

Hazırladığınız harçın üzerine streç filmle kapatarak yarı saat buz dolabında dinlendirin.

Ardından harçtan ceviz büyülüğünde parçalar koparıp yuvarlayın. 1 cm olacak şekilde üzerine bastırarak yassılaştırın.



Icon Oluşturma



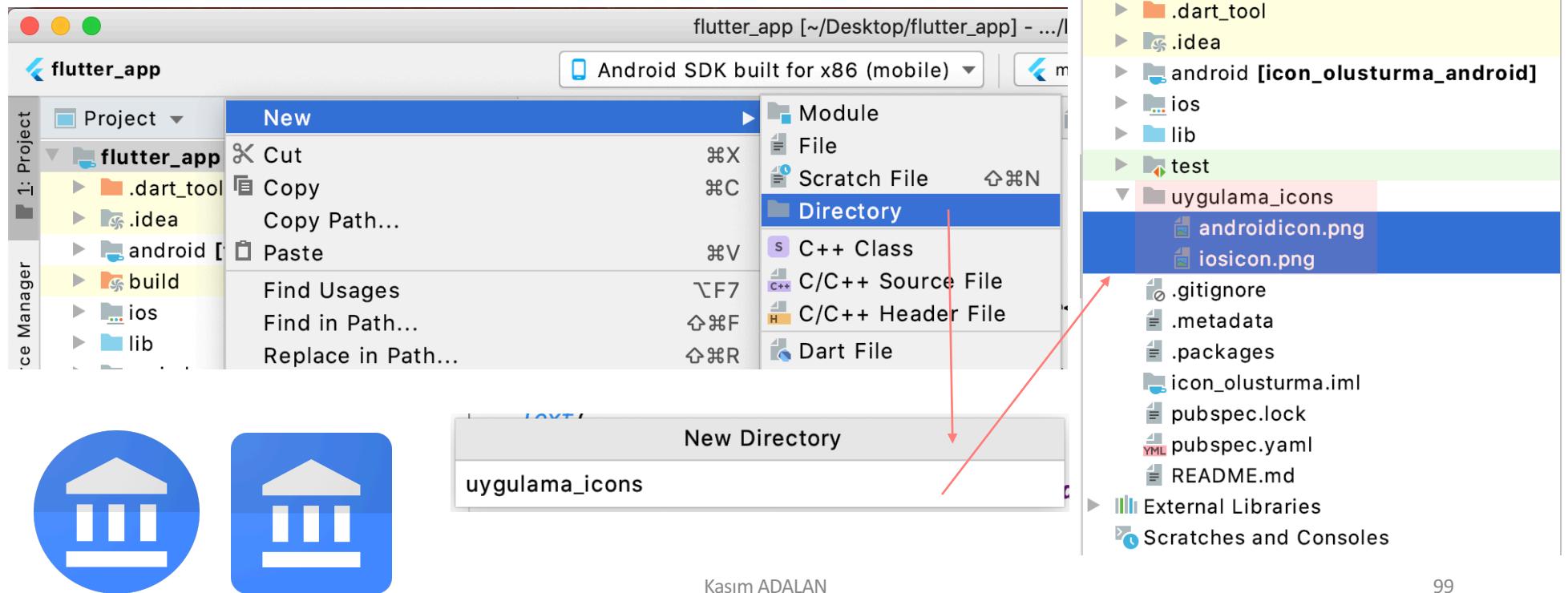
Icon Oluşturma



- Hem android hem ios için ayrı ayrı icon oluşturulabilir.
- Projeye icon resimleri eklenir.
- Bu resimler hem android hem ios için çeşitli ölçeklerde otomatik olarak oluşturulacaktır.
- Resimler tekrar çeşitli ölçeklerde boyutlandırılacağı için projeye eklenecek resim boyutu büyük olması iyi olacaktır.
- Android için minimum : 512x512 , IOS için minimum : 1024x1024 idealdir.
- Bu boyutlar her iki platform için kullanılan en büyük boyuttur.
- Bundan dolayı bu boyutlarda resimler eklemek daha verimli olacaktır.

Resimleri Ekleme

- Resimler için dosya oluşturulur.



Proje Kurulum

- **pubspec.yaml** dosyası içine gerekli sdk ve icon tanımlamaları yapılır.

```
main.dart x pubspec.yaml x
Flutter commands
Pub get Pub upgrade Pub outdated | Flutter doctor

32 dev_dependencies:
33   flutter_test:
34     sdk: flutter
35
36   flutter_launcher_icons: "^0.7.0" sdk kurulumu
37
38 flutter_icons:
39   ios: true platformlara izin verme
40   android: true
41   image_path: "uygulama_icons/iosicon.png"
42   image_path_android: "uygulama_icons/androidicon.png"

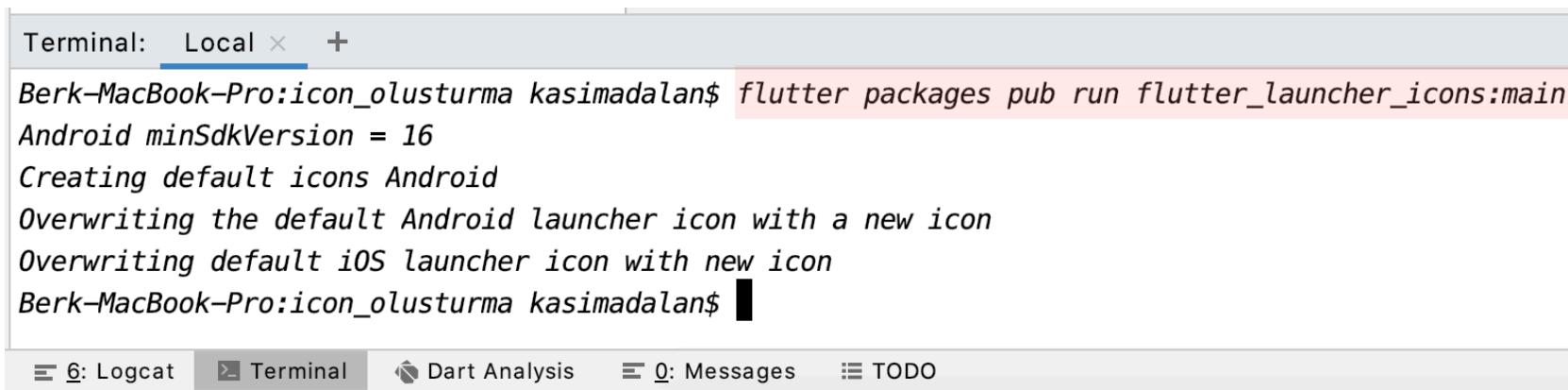
flutter_icons tanimlamasi en sola bitisik olmalidir.
```

- Tanımlamalar bitince Pub get çalıştırılır ve tanımlamalar yüklenir.

```
Messages: [icon_olusturma] Flutter x
flutter pub get
Running "flutter pub get" in icon_olusturma...
Process finished with exit code 0
1,7s
```

Proje Kurulumu

- Terminal ekranından iconların projeye eklenip çeşitli boyutlarda oluşmasını sağlamalıyız.
- Terminal ekranına **flutter packages pub run flutter_launcher_icons:main** yazmalıyız.



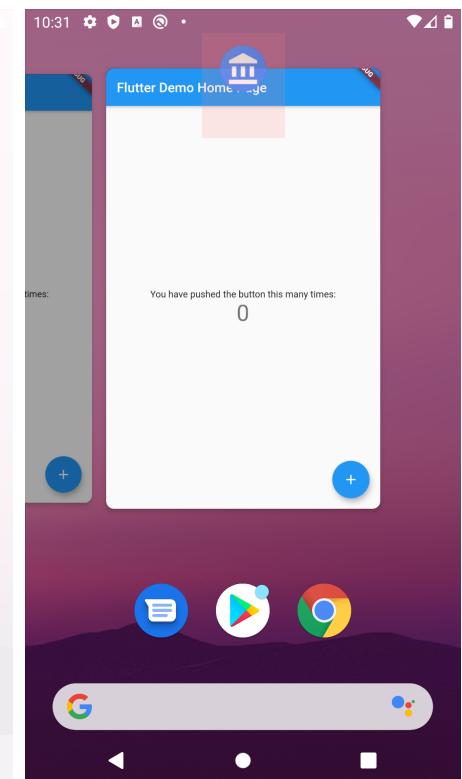
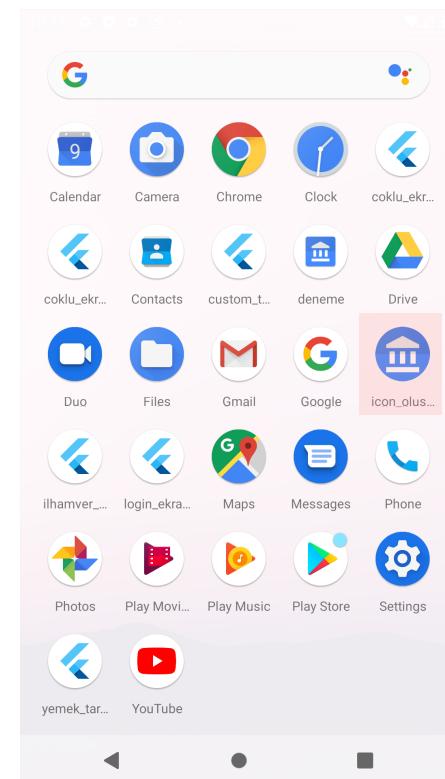
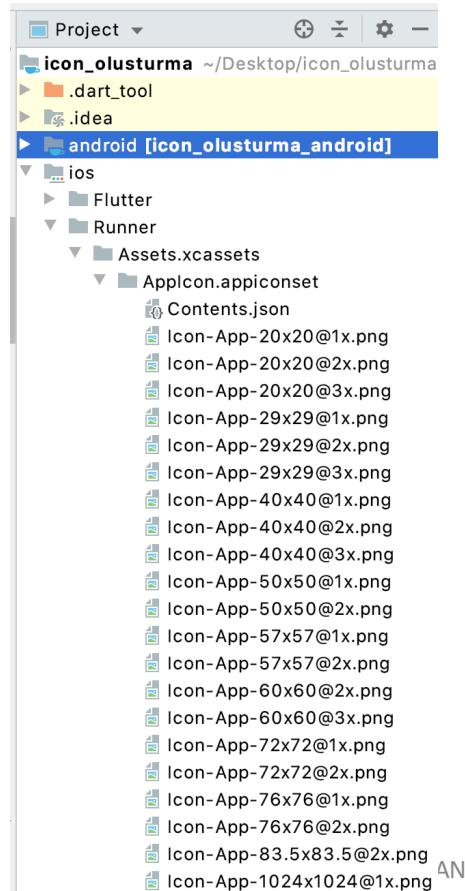
Terminal: Local +

```
Berk-MacBook-Pro:icon_olusturma kasimadalan$ flutter packages pub run flutter_launcher_icons:main
Android minSdkVersion = 16
Creating default icons Android
Overwriting the default Android launcher icon with a new icon
Overwriting default iOS launcher icon with new icon
Berk-MacBook-Pro:icon_olusturma kasimadalan$ █
```

6: Logcat Terminal Dart Analysis 0: Messages TODO

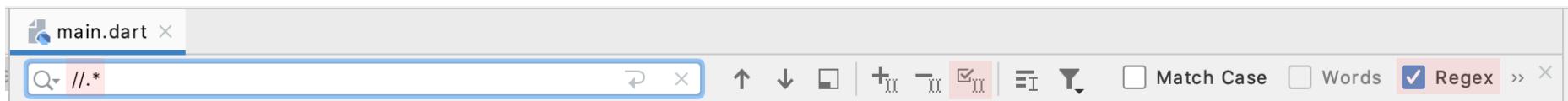
Icon Oluşum Kontrolü

- Proje içinde iconlarımız çeşitli boyutlarda oluşturulmaktadır.



Proje içindeki bütün yorumları silme

- Cntrl + F ile arama çubuğu açılır.
- `//.*` yazılır.
- Regex seçilir.
- Ardından bütün satırlar seçilir.
- Silme tuşuna basılarak bütün yorumlar silinir.



Teşekkürler...



kasim-adalan



kasimadalan@gmail.com



kasimadalan