

Mobil Programlama ve Tasarım

Hafta 3

Özel Widget Oluşturma - Class

```
31 }  
32  
33 class GradientContainer {  
34  
35 }
```

```
32  
33 class GradientContainer extends StatelessWidget {  
34   @override  
35   Widget build(context) {  
36     return ;  
37   }  
38 }
```

```
33 class GradientContainer extends StatelessWidget {  
34  
35 }
```

```
33 class GradientContainer extends StatelessWidget {  
34   build()  
35 }
```

```
class GradientContainer extends StatelessWidget {  
  @override  
  Widget build() {}  
}
```

Ve return yanına container widget
Ve içeriğini ekliyor, body widget'ında
Bu class'ı çağırıyoruz.

İyileştirmeler

```
class GradientContainer extends StatelessWidget {  
  const GradientContainer({super.key});  
  
  @override  
  Widget build(context) {
```

```
4 runApp(  
5   ⚡ const MaterialApp(  
6     home: Scaffold(  
7       body: GradientContainer(),  
8     ), // Scaffold  
9   ), // MaterialApp  
10 );  
11 }  
  
12  
13 class GradientContainer extends StatelessWidget {  
14   const GradientContainer({super.key});  
15 }
```

Kodları Dosyalara Ayıralım

```
main.dart X gradient_container.dart

lib > main.dart > main
1 import 'package:flutter/material.dart';
2 import 'package:uyg_1/gradient_container.dart';
3
Run | Debug | Profile
4 void main() {
5   runApp(
6     const MaterialApp(
7       //düzenleme
8       home: Scaffold(
9         body: GradientContainer(),
10      ), // Scaffold
11    ), // MaterialApp
12  );
13 }
```

```
main.dart X gradient_container.dart X

lib > gradient_container.dart > ...
1 import 'package:flutter/material.dart';
2
3 class GradientContainer extends StatelessWidget {
4   const GradientContainer({super.key}); //düzenleme
5   @override
6   Widget build(context) {
7     return Container(
8       decoration: const BoxDecoration(
9         gradient: LinearGradient(
10          colors: [
11            Color.fromARGB(255, 15, 153, 187),
12            Colors.white,
13          ],
14        ), // LinearGradient
15      ), // BoxDecoration
16      child: const Center(
17        child: Text(
18          'Merhaba Dünya',
19          style: TextStyle(color: Colors.black, fontSize: 28),
20        ), // Text
21      ), // Center
22    ); // Container
23  }
24 }
```

Uygulama

- Örnekte kullandığımız Text için başka bir Dart sayfasında yeni bir sınıf oluşturun ve ilgili sayfa ile bağlantısını yapın.

lib > gradient_cnt.dart > ...

```
1 import 'package:flutter/material.dart';
2 import 'package:uyg1_bm/text_sinif.dart';
3
4 class GradientContainer extends StatelessWidget {
5   const GradientContainer({super.key});
6   @override
7   Widget build(context) {
8     return Container(
9       decoration: const BoxDecoration(
10         gradient: LinearGradient(
11           colors: [
12             Color.fromARGB(255, 76, 175, 168),
13             Colors.white,
14           ],
15         ), // LinearGradient // BoxDecoration
16       child: const Center(
17         child: TextSnf(),
18       ), // Center
19     ); // Container
20   }
21 }
```

main.dart gradient_cnt.dart text_sinif.dart X

lib > text_sinif.dart > TextSnf

```
1 import 'package:flutter/material.dart';
2
3 class TextSnf extends StatelessWidget {
4   const TextSnf({super.key});
5   @override
6   Widget build(context) {
7     return const Text(
8       'Merhaba Dünya!',
9       style: TextStyle(
10         color: Color.fromARGB(255, 93, 54, 54),
11         fontSize: 28,
12       ), // TextStyle
13     ); // Text
14   }
15 }
```

Class'ı daha
kullanışlı
hale getirme

```
lib > text_sinif.dart > TextSnf > build
1  import 'package:flutter/material.dart';
2
3  class TextSnf extends StatelessWidget {
4      const TextSnf({super.key});
5      @override
6      Widget build(context) {
7          return const Text(
8              'Merhaba Dünya!',
9              style: TextStyle(
10                 color: Color.fromARGB(255, 93, 54, 54),
11                 fontSize: 28,
12             ), // TextStyle
13          ); // Text
14      }
15  }
```


Olması beklenen,
bu class'ta text
bölümüne
dışarıdan yazı
girebilmek ve
tekrar
kullanabilmek

```
lib > gradient_cnt.dart > GradientContainer > build
1  import 'package:flutter/material.dart';
2  import 'package:uyg1_bm/text_sinif.dart';
3
4  class GradientContainer extends StatelessWidget {
5    const GradientContainer({super.key});
6    @override
7    Widget build(context) {
8      return Container(
9        decoration: const BoxDecoration(
10          gradient: LinearGradient(
11            colors: [
12              Color.fromARGB(255, 76, 175, 168),
13              Colors.white,
14            ],
15          ), // LinearGradient // BoxDecoration
16          child: const Center(
17            child: TextSnf('Merhaba Dünya!'),
18          ), // Center
19        ); // Container
20    }
21  }
```

Bunu oluşturabilmek için `const TextSnf`'ın metinsel değer alabileceğini belirtmeliyiz

```
3 class TextSnf extends StatelessWidget {  
4   const TextSnf(String text, {super.key});  
5   @override  
6   Widget build(context) {
```

Merhaba Dünya yazısını bu classtan sildiğimizde ve bir text'in yazdırılmasını istediğimizde hata alacağız

```
3 class TextSnf extends StatelessWidget {  
4   const TextSnf(String text, {super.key});  
5  
6   @override  
7   Widget build(context) {  
8     return const Text(  
9       text;  
10      style: TextStyle(  
11        color: Color.fromARGB(255, 93, 54, 54),  
12        fontSize: 28,  
13      ),  
14    );  
15  }  
16 }
```

Bir değişken tanımlamamız ve bunu class'ta çağırmamız gerekecek

```
2
3  ✓ class TextSnf extends StatelessWidget {
4      const TextSnf(this.text, {super.key});
5
6      String text;
7
8      @override
9      ✓ Widget build(context) {
10     ✓     return const Text(
11         text;
12         style: TextStyle(
```

Artık const
TextSnf'in const
olmaması ve
return'deki Text'in
de const olmaması
gerekıyor, TextSyle
ise const olmalı

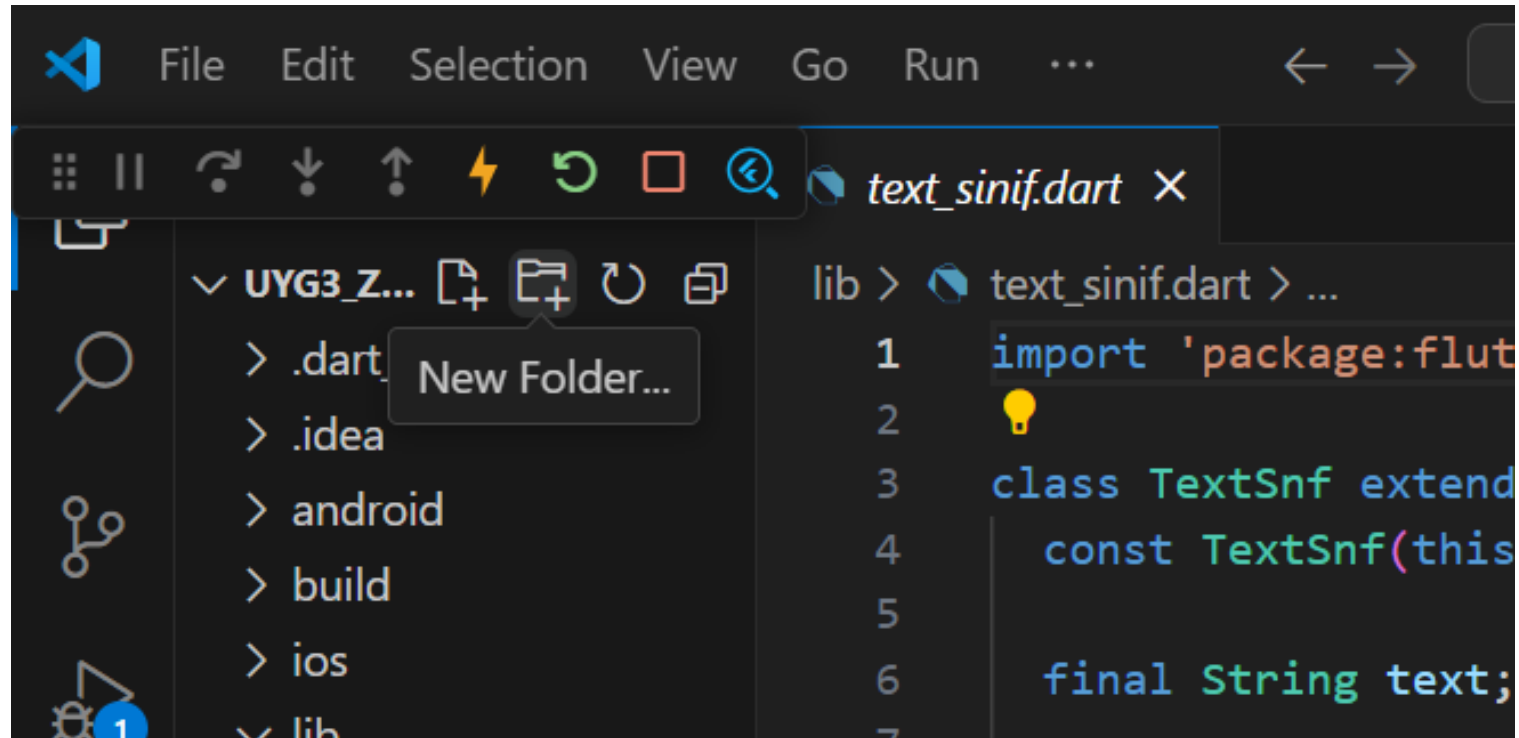
```
lib > text_sinif.dart > TextSnf
1 import 'package:flutter/material.dart';
2
3 class TextSnf extends StatelessWidget {
4   TextSnf(this.text, {super.key});
5
6   String text;
7
8   @override
9   Widget build(context) {
10     return Text(
11       text,
12       style: const TextStyle(
13         color: Color.fromARGB(255, 93, 54, 54),
14         fontSize: 28,
15       ), // TextStyle
16     ); // Text
17   }
18 }
```

Burada aslında text değeri uygulama çalışırken değiştirilmeyecek şekilde düzenlendiği için daha düzenli bir hale getirebilir `final` türünde tanımlabiliriz.

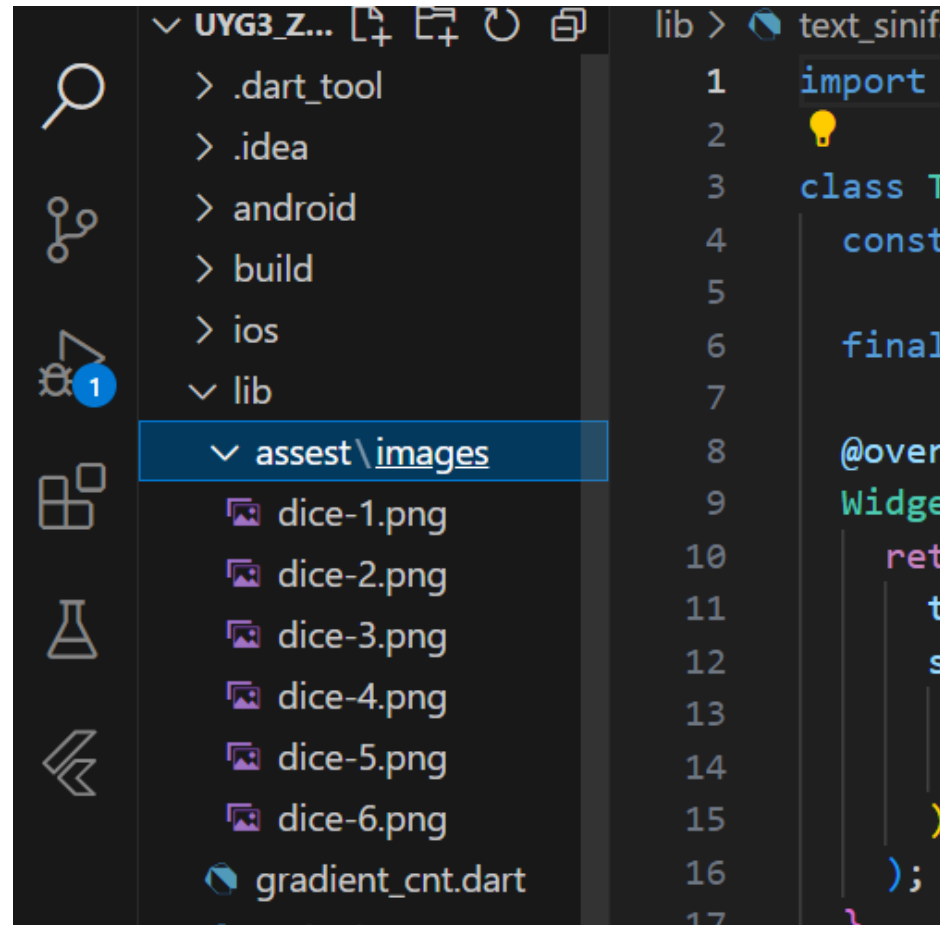
```
3 class TextSnf extends StatelessWidget {  
4   const TextSnf(this.text, {super.key});  
5  
6   final String text;  
7  
8   @override  
9   Widget build(context) {  
10    return Text(  
11      text,  
12      style: const TextStyle(  
13        color: Color.fromARGB(255, 93, 54, 54),  
14        fontSize: 28,  
15      ), // TextStyle
```

Zar Oyunu

Kullanacağımız Görseller İçin Yeni Bir Klasör Oluşturma assest/images



Görsellerimizi Klasörümüze Ekliyoruz ve pubspec.yaml dosyamızda tanımlıyoruz.



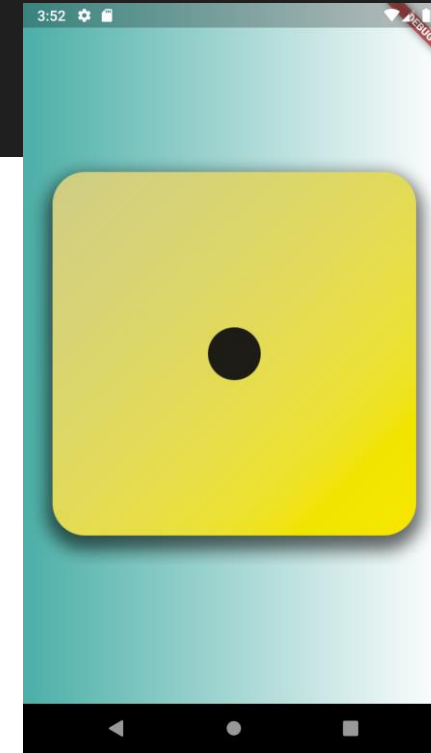

```
59 # the Material Icons class.
60 uses-material-design: true
61
62 # To add assets to your application, add an assets section, like this:
63 # assets:
64 #   - images/a_dot_burr.jpeg
65 #   - images/a_dot_ham.jpeg
66
67 # An image asset can refer to one or more resolution-specific "variants", see
68 # https://flutter.dev/assets-and-images/#resolution-aware
69
70 # For details regarding adding assets from package dependencies, see
71 # https://flutter.dev/assets-and-images/#from-packages
72
```

```
60 uses-material-design: true
61
62 # To add assets to your application, add an assets section, like this:
63 assets:
64   - lib/assent/images/dice-1.png
65   - lib/assent/images/dice-2.png
66   - lib/assent/images/dice-3.png
67   - lib/assent/images/dice-4.png
68   - lib/assent/images/dice-5.png
69   - lib/assent/images/dice-6.png
70
```

GradientContainer sınıfında text yerine bu görselleri çağıralım

```
4 class GradientContainer extends StatelessWidget {  
5   const GradientContainer({super.key});  
6   @override  
7   Widget build(context) {  
8     return Container(  
9       decoration: const BoxDecoration(  
10        gradient: LinearGradient(  
11         colors: [  
12           Color.fromARGB(255, 76, 175, 168),  
13           Colors.white,  
14         ],  
15       ), // LinearGradient // BoxDecoration  
16       child: const Center(  
17         child: TextSnf('Merhaba Dünya!'),  
18       ), // Center  
19     ); // Container  
20   }  
21 }
```

```
Colors.white,  
],  
)), // LinearGradient // BoxDecoration  
child: Center(  
  child: Image.asset('lib/assest/images/dice-1.png'),  
), // Center  
); // Container  
}  
}
```



```
5      )), // LinearGradient // BoxDecoration
6      child: Center(
7        child: Image.asset(
8          'lib/assest/images/dice-1.png',
9          width: 200,
10         ), // Image.asset
11       ), // Center
12     ); // Container
13   }
```



Column oluştururak buna zar ve bir de buton ekleyeceğiz

```
16  child: Center(  
17  child: Column(children: [],) Image.asset(  
18    'lib/assest/images/dice-1.png',  
19    width: 200,  
20  ), // Image.asset  
21  ), // Center  
22  ); // Container  
23  }  
24  }
```

```
    ), // LinearGradient // BoxDecoration  
    child: Center(  
      child: Column(  
        children: [  
          Image.asset(  
            'lib/assest/images/dice-1.png',  
            width: 200,  
          ), // Image.asset  
        ],  
      ), // Column  
    ), // Center  
  ); // Container
```

Şimdi Buton ekleyelim

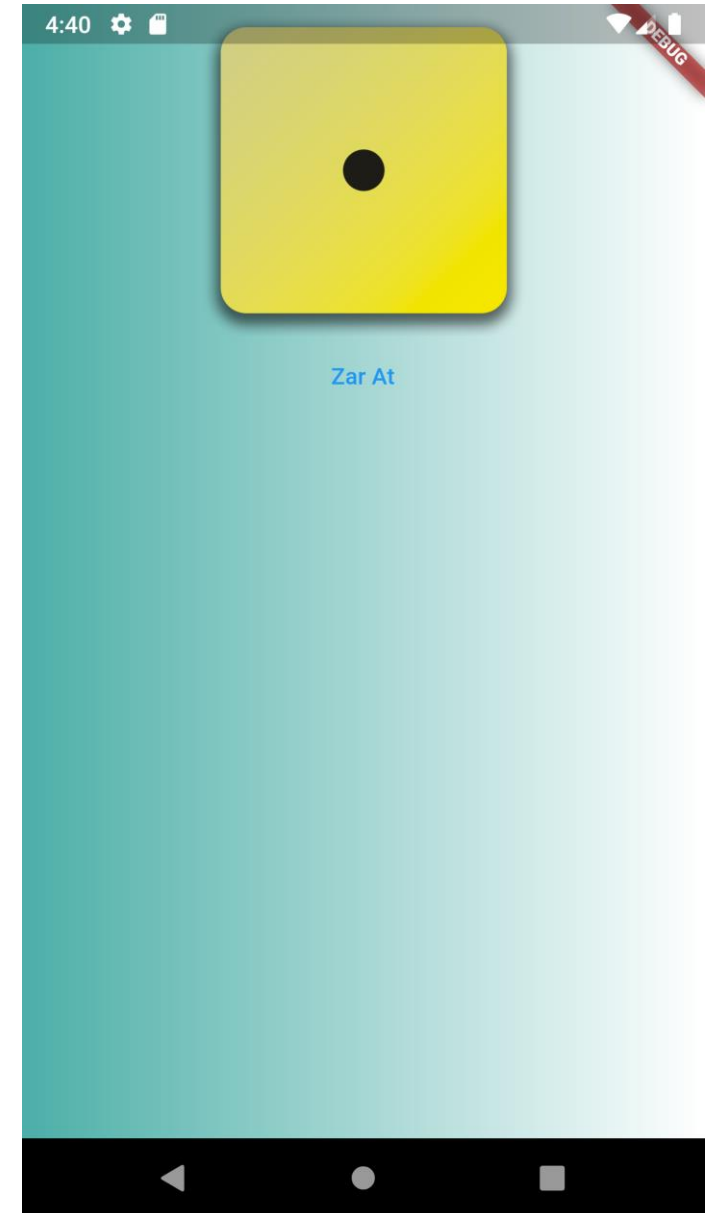
```
child: Center(  
  child: Column(  
    children: [  
      Image.asset(  
        'lib/assest/images/dice-1.png',  
        width: 200,  
      ), // Image.asset  
      TextButton(onPressed: onPressed, child: child)  
    ],  
  ), // Column  
) // Center
```

```
), // Image.asset  
TextButton(onPressed: onPressed, child: const Text('Zar At'),)
```

onPressed için bir fonksiyon yazmamız ve burada çağırmamız gerekiyor.

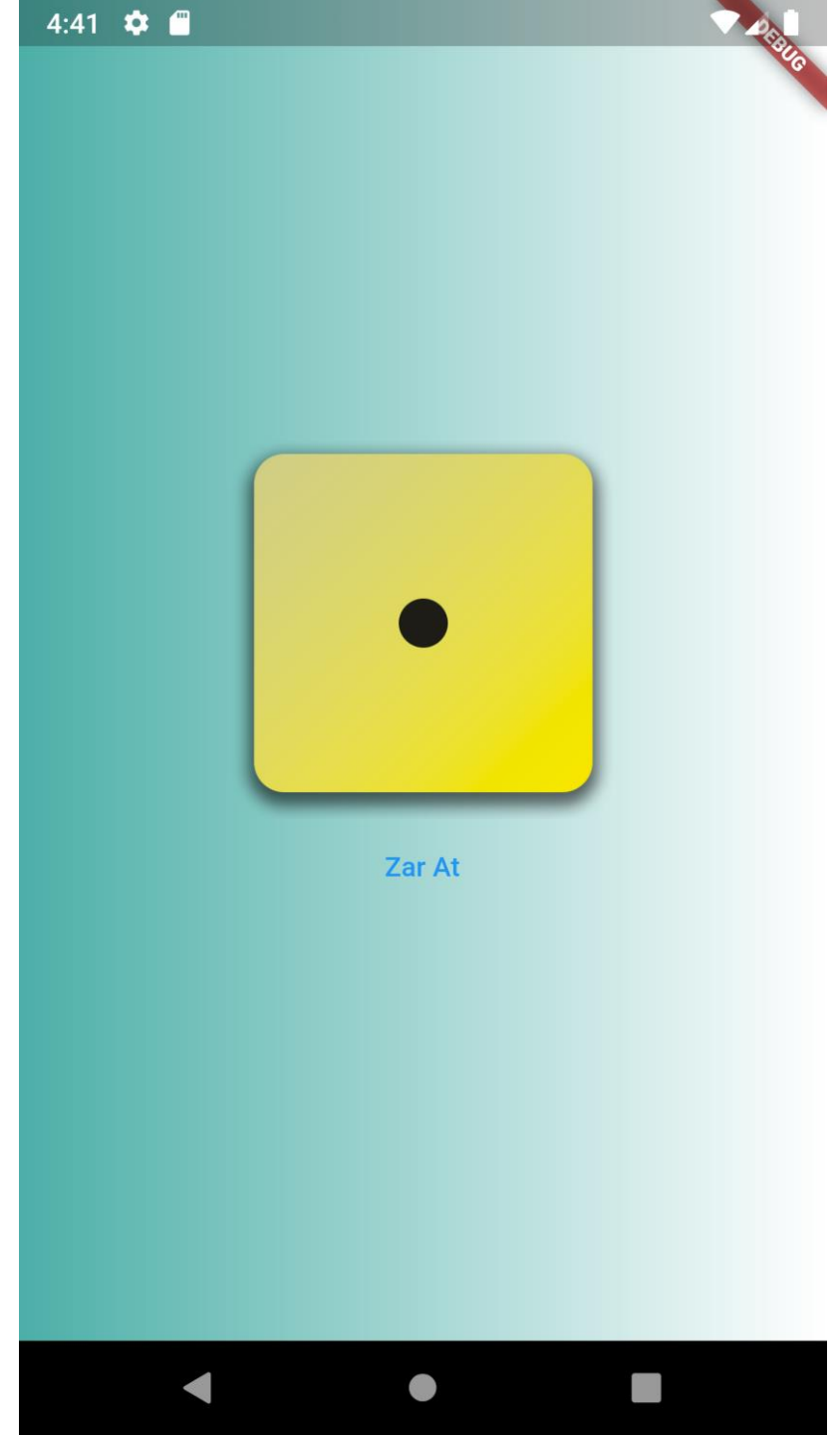
```
2 //import package:uygi_b  
3   
4 void zarAt() {  
5   //....  
6 }  
7   
8 class GradientContainer  
9   const GradientContaine  
10  @override
```

```
), // Image.asset  
TextButton(  
  onPressed: zarAt,  
  child: const Text('Zar At'),  
) // TextButton  
],  
), // Column  
) // Center
```



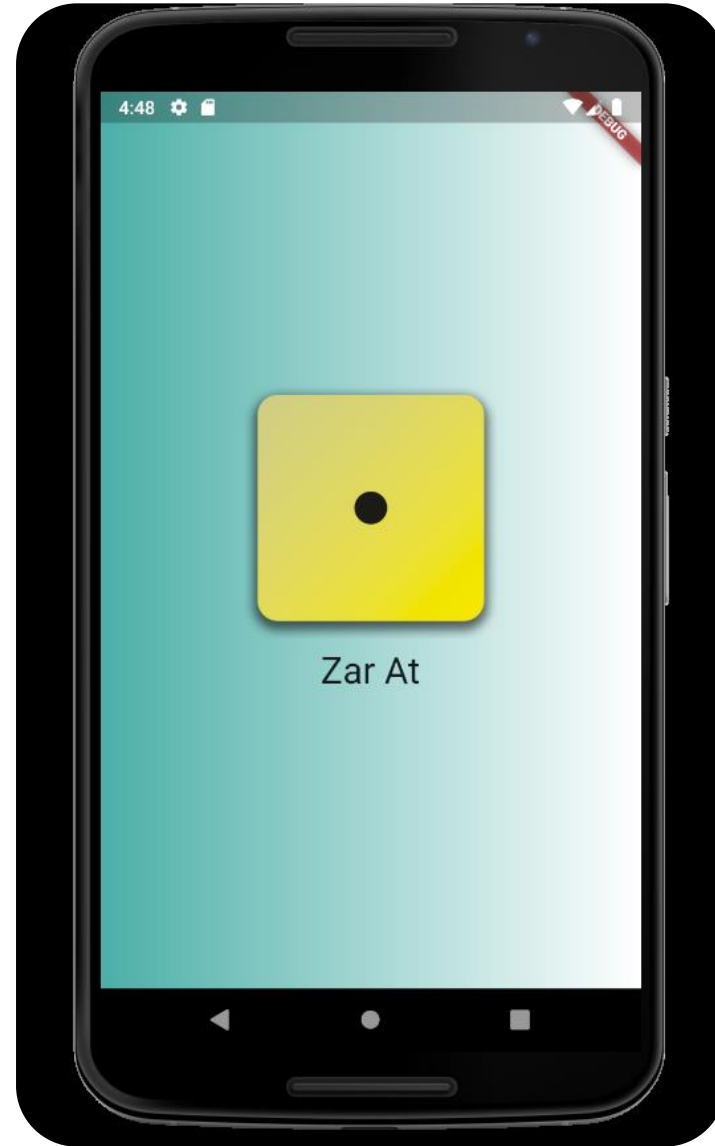
Column içeriğini ortalayalım

```
9      )), // LinearGradient // BoxDecoration
10     child: Center(
11       child: Column(
12         mainAxisAlignment: MainAxisAlignment.min,
13         children: [
14           Image.asset(
15             'lib/assest/images/dice-1.png',
16             width: 200,
17           ), // Image.asset
18           const TextButton(
19             onPressed: zarAt,
```



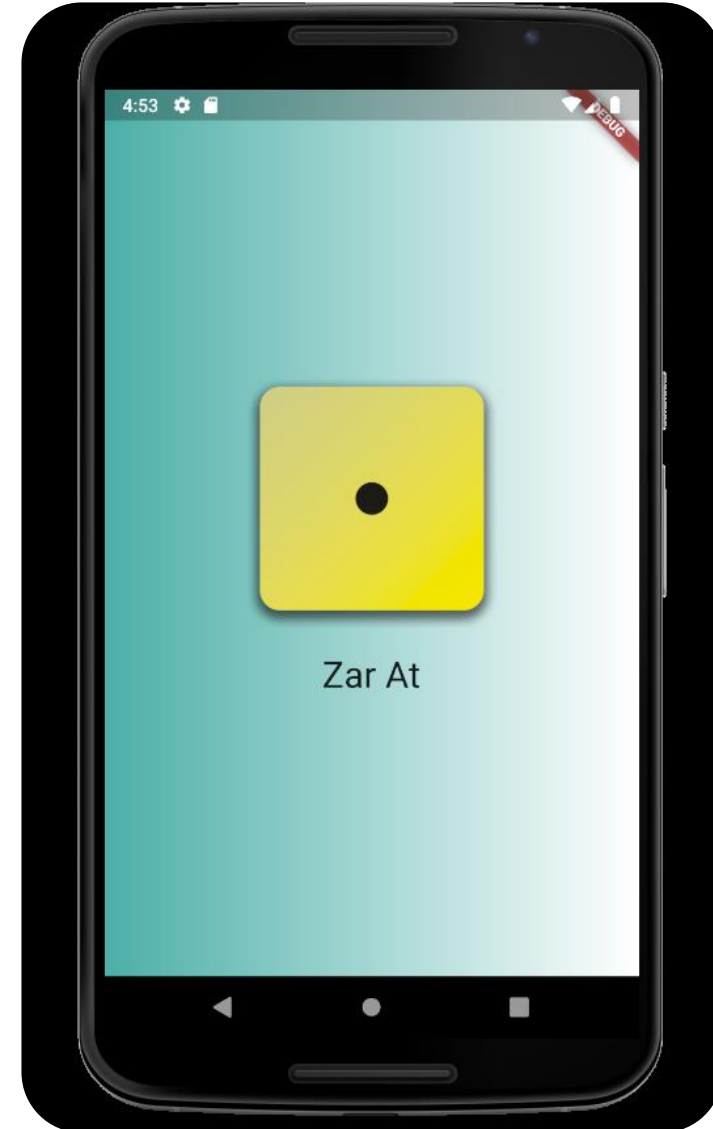
Şimdi Butonunun Stilini Düzenleyelim

```
width: 200,  
) , // Image.asset  
TextButton(  
  onPressed: zarAt,  
  style: TextButton.styleFrom(  
    foregroundColor: Colors.black87,  
    textStyle: const TextStyle(  
      fontSize: 28,  
    ), // TextStyle  
  ),  
  child: const Text('Zar At'),  
) // TextButton  
],
```



Botun ile Görsel Arasındaki Boşluğu Düzenleme

```
), // Image.asset  
TextButton(  
  onPressed: zarAt,  
  style: TextButton.styleFrom(  
    padding: const EdgeInsets.only(top:20,),  
    foregroundColor: Colors.black87,  
    textStyle: const TextStyle(  
      fontSize: 28,  
    ) // TextStyle
```



Butonun İşlevini Düzenleme ve Görsellerin Değişmesini Sağlama

```
4
5
6 class GradientContainer extends StatelessWidget {
7   const GradientContainer({super.key});
8
9   var aktifZarGorseli = 'lib/assest/images/dice-1.png';
10
11 void zarAt() {
12   //....
13 }
```

```
child: Center(
  child: Column(
    mainAxisAlignment: MainAxisAlignment.min,
    children: [
      Image.asset(
        aktifZarGorseli,
        width: 200,
      ), // Image.asset
      TextButton(
        onPressed: zarAt,
```

```
6 class GradientContainer extends StatelessWidget {
7   GradientContainer({super.key});
8
9   var aktifZarGorseli = 'lib/assest/images/dice-1.png';
10
11 void zarAt() {
12   //....
13 }
```

Butona Tıkladığımızda 2. Görselin Gelmesini İstedığımızı varsayarak fonksiyonu yeniden hazırlayalım

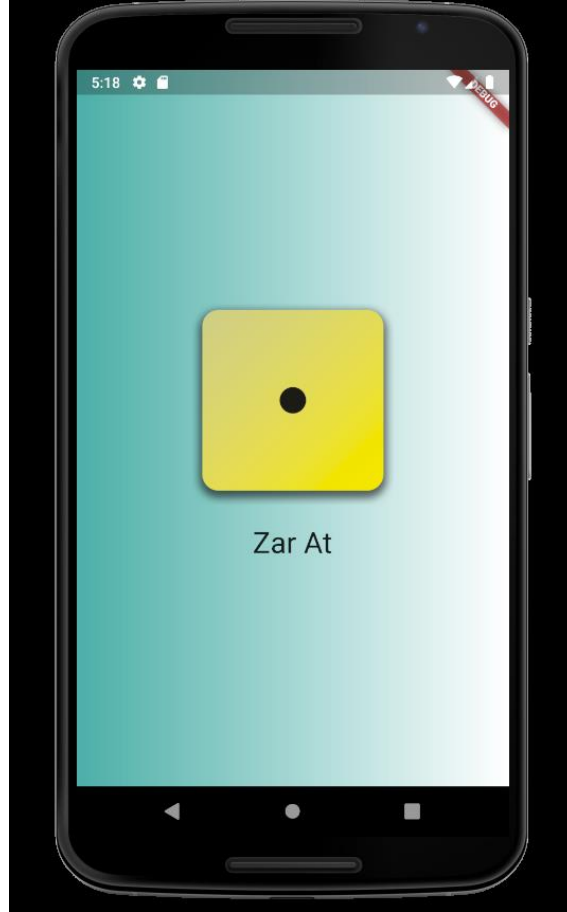
```
4  class GradientContainer extends StatelessWidget {  
5      GradientContainer({super.key});  
6  
7      var aktifZarGorseli = 'lib/assest/images/dice-1.png';  
8  
9      void zarAt() {  
10         aktifZarGorseli = 'lib/assest/images/dice-2.png';  
11     }  
12  
13
```

Main.dart'da bu classı çağırdığımız için düzenleme yapmalıyız

```
lib > main.dart > main
1  import 'package:flutter/material.dart';
2  import 'package:uyg1_bm/gradient_cnt.dart';
3
4  void main() {
5    runApp(
6      const MaterialApp(
7        home: Scaffold(
8          body: GradientContainer(),
9        ), // Scaffold
10      ), // MaterialApp
11    );
12  }
13
```

```
lib > main.dart > main
1  import 'package:flutter/material.dart';
2  import 'package:uyg1_bm/gradient_cnt.dart';
3
4  void main() {
5    runApp(
6      MaterialApp(
7        home: Scaffold(
8          body: GradientContainer(),
9        ), // Scaffold
10      ), // MaterialApp
11    );
12  }
13
```

Botuna Tıklayıp Denediğimizde Görselin
Değişmediğini Görüyoruz.



StatelessWidget

- Bu widget türü çalışma zamanında değişmeyecek olan sınıflarda kullanılabilir. Uygulama içinde (bizim hazırladığımız zar uygulamasında olduğu gibi) görsel, yazı, renk v.b. özellikler kullanıcı etkileşimi ile değiştiğinde bu widgetın kullanılması uygun değildir. Yerine **StatefulWidget** kullanılmalıdır.

Düzenleme

- Bu durumda değişecek olan verilerin bulunduğu widgetların olduğu bir `StatefulWidget` sınıfı değildir.
- Bir de değişmeyen bölümler için `StatelessWidget` bulunmalıdır.
- Bunun için zar atma işlemlerinde değişen bölümler olduğu için bu kısımları `StatefulWidget` içinde başka bir sayfada tanımlayacağız.

StatefulWidget

```
lib > dice_roller.dart > _DiceRollerState
1  import 'package:flutter/material.dart';
2
3  class DiceRoller extends StatefulWidget{
4  @override
5  State<DiceRoller> createState() {
6  |   return
7  | }
8
9  }
10
11 class DiceRollerState
```

```
lib > dice_roller.dart > _DiceRollerState
1  import 'package:flutter/material.dart';
2
3  class DiceRoller extends StatefulWidget{
4  @override
5  State<DiceRoller> createState() {
6  |   return
7  | }
8
9  }
10
11 class DiceRollerState extends State<DiceRoller>{
12 |   build(){}
13 }
```

```
10
11 class DiceRollerState extends State<DiceRoller>{
12 |   @override
13 |   Widget build(context){
14 |   |   return
15 |   | }
16 }
```

StatefulWidget

```
9 }
10
11 class _DiceRollerState extends State<DiceRoller>{
12   @override
13   Widget build(context){
14     return // bu bölüme column kısmı gelmeli
15   }
16 }
```

```
class _DiceRollerState extends State<DiceRoller>{
  @override
  Widget build(context){
    return Column(
      mainAxisAlignment: MainAxisAlignment.min,
      children: [
        Image.asset(
          aktifZarGorseli,
          width: 200,
        ), // Image.asset
        TextButton(
          onPressed: zarAt,
          style: TextButton.styleFrom(
            padding: const EdgeInsets.only(
              top: 20,
            ), // EdgeInsets.only
            foregroundColor: Colors.black87,
            textStyle: const TextStyle(
```

NOT: Eklediğimiz Column'un son satırındaki , işaretini ; olarak değiştirmeyi unutmayın.

StatefulWidget

```
1 import 'package:flutter/material.dart';
2
3 class DiceRoller extends StatefulWidget {
4   @override
5   State<DiceRoller> createState() {
6     return _DiceRollerState();
7   }
8 }
9
10 class _DiceRollerState extends State<DiceRoller> {
11   @override
12   Widget build(context) {
13     return Column(
14       mainAxisAlignment: MainAxisAlignment.min,
15       children: [
16         Image.asset(
17           aktifZarGorseli,
18           width: 200,
19         ), // Image.asset
20         TextButton(
```

```
2
3 class DiceRoller extends StatefulWidget {
4   const DiceRoller({super.key});
5   @override
6   State<DiceRoller> createState() {
7     return _DiceRollerState();
8   }
9 }
10
11 class _DiceRollerState extends State<DiceRoller> {
12   @override
13   Widget build(context) {
14     return Column(
15       mainAxisAlignment: MainAxisAlignment.min,
16       children: [
17         Image.asset(
18           aktifZarGorseli,
19           width: 200,
20         ), // Image.asset
21         TextButton(
```

Bazı eksikleri düzenleme zamanı

```
children: [  
  Image.asset(  
    aktifZarGorseli,  
    width: 200,  
  ), // Image.asset  
  TextButton(  
    onPressed: zarAt,  
    style: TextButton.styleFrom(  
      padding: const EdgeInsets.only(  
        top: 20,  
      ) // EdgeInsets only
```

Değişken ve fonksiyonları **StatefulWidget** sınıfını kullandığımız sayfaya alıyoruz

```
3
4 class GradientContainer extends StatelessWidget {
5   GradientContainer({super.key});
6
7   var aktifZarGorseli = 'lib/assest/images/dice-1.png';
8
9   void zarAt() {
10    aktifZarGorseli = 'lib/assest/images/dice-2.png';
11  }
12
13  @override
```

Diğer sayfalarda düzenlemeler

```
Flutter (Copy2Nexus 6 A) | dice_roller.dart | main.dart
lib > main.dart > main
1 import 'package:flutter/material.dart';
2 import 'package:uyg1_bm/gradient_cnt.dart';
3
Run | Debug | Profile
4 void main() {
5   runApp(
6     const MaterialApp(
7       home: Scaffold(
8         body: GradientContainer(),
9       ), // Scaffold
10     ), // MaterialApp
11   );
12 }
13
```

```
lib > gradient_cnt.dart > GradientContainer > GradientContainer
1 import 'package:flutter/material.dart';
2 //import 'package:uyg1_bm/text_sinif.dart';
3
4 class GradientContainer extends StatelessWidget {
5   const GradientContainer({super.key});
6
7   @override
8   Widget build(context) {
9     return Container(
10       decoration: const BoxDecoration(
11         gradient: LinearGradient(
12           colors: [
13             Color.fromARGB(255, 76, 175, 168),
14
```

DiceRoller'u Gradient sayfamızda çağırıyoruz ve sayfayı import ediyoruz

```
lib > gradient_cnt.dart > GradientContainer > build
5 class GradientContainer extends StatelessWidget {
8   Widget build(context) {
9     return Container(
10      decoration: const BoxDecoration(
11        gradient: LinearGradient(
12          colors: [
13            Color.fromARGB(255, 76, 175, 168),
14            Colors.white,
15          ],
16        ), // LinearGradient // BoxDecoration
17      child: Center(
18        child: DiceRoller(),
19      ), // Center
20    ); // Container
21  }
22 }
```

Tekrar çalıştırdığımızda build metodu içerisinde değişiklik olmasını sağlamış olmayacağız. Bu sebeple özel bir fonksiyon kullanmamız gerekecek ve ZarAt fonksiyonunu düzenleyeceğiz.

```
✓ class _DiceRollerState extends State<DiceRoller> {  
  var aktifZarGorseli = 'lib/assest/images/dice-1.png';  
  
  void zarAt() {  
    setState(() {  
      aktifZarGorseli = 'lib/assest/images/dice-3.png';  
    });  
  }  
}
```

Rastgele gelen sayıya göre zar görselinin değişmesi

```
lib > dice_roller.dart > ...  
1 import 'package:flutter/material.dart';  
2 import 'dart:math';  
3  
4 class DiceRoller extends StatefulWidget {  
5   constructor({super.key}) {
```

```
void zarAt() {  
  Random().nextInt(6) + 1;  
  setState(() {  
    aktifZarGorseli = 'lib/assest/images/dice-3.png';  
  });  
}
```

```
void zarAt() {  
  var zarNo= Random().nextInt(6) + 1;  
  setState(() {  
    aktifZarGorseli = 'lib/assest/images/dice-$zarNo.png';  
  });  
}
```