

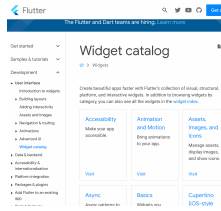
# Görsel Programlama

## Sık kullanılan Widget'lar

Emir ÖZTÜRK

## Widget Catalog

- Tüm widget'ların listesi
- <https://docs.flutter.dev/development/ui/widgets>



## Expanded

- child
- flex
- Birden fazla expanded için flex'ler arası oran

Expanded ile bir pencerenin flex'lerin toplamı kadar eşit parçaya bölünüp boyutlandırılması sağlanabilmektedir. Responsive tasarım için her pencerenin ne kadar yer kaplayacağını belirtmesi için kullanılabilir.

## Image

- Image(image:)
- Image.new
- Image.asset
- Image.network
- Image.file
- Image.memory
- Desteklenen formatlar
  - JPEG, PNG, GIF, Animated GIF, WebP, Animated WebP, BMP

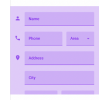


Image  
A widget that displays an image.

Ekrana resim eklemek için Image widget'ı kullanılabilir. Image widget'ı ağdan, bellekten veya bir dosyadan resim gösterme işlemi yapabilir. Bir çok dosya formatı desteklenmektedir.

## TextField

- Kullanıcıdan alınan girdi
- Stateful
- Özellikler
  - autocorrect -> bool
  - decoration -> InputDecoration
  - enabled -> bool
  - maxLength -> int
  - obscureText -> bool
  - textAlign -> TextAlign
- Olaylar
  - onChanged
  - onSubmitted
  - onTap



TextField  
Touching a text field places the cursor and displays the keyboard. The TextField widget implements this component.

Ekrandan girdi alabilmek için TextField kullanılır. Stateful widget içerisinde tanımlanmalıdır. TextField içerisindeki değer TextEditingController'a bağlanır. Bu controller state çizdirme durumlarında saklanmalıdır. TextField'ın içerisindeki metin değiştirildiğinde, tıklandığında veya yazma işlemi bittiğinde submit tuşuna basıldığında tetiklenecek metotları bulunmaktadır. Ayrıca oto düzeltme, maksimum uzunluk, şifreli gösterim gibi özellikleri de bulunmaktadır.

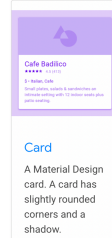
## RichText

- text
- style
- Children->TextSpan listesi
- maxLines -> int
- overflow -> TextOverflow
- softWrap -> bool
- textDirection -> TextDirection
- textAlign -> TextAlign

Birden fazla satıra sahip ve biçimlendirilmiş metin saklamak için RichText kullanılabilir. Maksimum satır sayısı, taşma durumu, satırın ekrana sığmadığında alt satıra geçip geçmeyeceği, text'in yönü gibi özellikleri atanabilir.

## Card

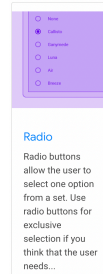
- Köşeli ve yükseltilmiş
- child
- color -> Color
- elevation -> double
- margin -> EdgeInsets
- shape -> ShapeBorder



Uygulamalarda belirli grupların listelenmesi için Card bileşeninin kullanımı oldukça uygundur. Örneğin bir satış sitesinin ürünleri veya bir blog'un yazılarının önizlemelerinin bulunduğu kartlar tanımlamak için Card bileşeni kullanılabilir. Card bileşeninde Renk, boşluk, kartın şekli ve kartın yüksekliği (gölgelendirme ile) belirlenebilir.

## Radio

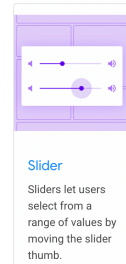
- Çok seçenekte tek seçim
- Özellikler
  - value
  - fillColor
- Olaylar
  - onChanged



Bir arayüzde birden fazla seçeneğin içerisinde bir seçeneğin seçilmesi için Radio kullanılabilir. Soru cevapları, anketler gibi seçimlerde kullanılması örnek verilebilir. Seçimin değişmesi durumunda tetiklenecek olay onChanged olarak belirlenir.

## Slider

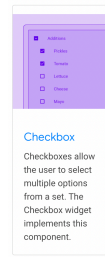
- Özellikler
  - divisions -> int
  - label -> String
  - max -> double
  - min -> double
  - value -> double
- Olaylar
  - onChanged
  - onChangedEnd



Slider ile belirli değer aralıklarında seçim yapılacak durumlar karşılanabilir. Örneğin sesin kısılıp açılması, ekran parlaklığının ayarlanması gibi arayüz bileşenleri için slider kullanımı uygundur. Slider değiştiğinde ve değişim işlemi bitirildiğinde tetiklenmek üzere iki farklı olayı bulunmaktadır. Ayrıca slider'ın minimum, maksimum ve atlama değerleri verilebilir.

## Checkbox

- Birden çok seçenek için bir veya daha fazla seçim
- Özellikler
  - checkColor -> Color
  - value



Radio bileşeninde çok seçenektan yalnızca biri seçilebilecekken Checkbox bileşeninde ise çoklu seçim mümkündür. Birden fazla doğru cevabı olan sorularda veya birden fazla seçilebilecek uygun seçeneklerde (Örneğin sevilen müzik türlerinin seçimi, ilgi alanlarının seçimi gibi) kullanım alanı bulunmaktadır. Radio bileşeninde olduğu gibi value ile seçilip seçilmediği tespit edilebilir.

## GridView

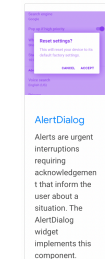
- GridView.count()
- children
- Padding
- mainAxisAlignment
- crossAxisAlignment
- crossAxisCount



Elemanların yan yana veya alt alta listelenmesinde Row ve Column kullanılırken, matris şeklinde bir kullanım için GridView kullanılabilir. Count ile eleman sayısı alınabilir, children ile elemanlar gride eklenebilir. Grid elemanları arasında boşluk vermek için main ve cross axis spacing ve elemanların her satırda kaç tane bulunacağını belirtmek için ise crossaxiscount kullanılabilir.

## AlertDialog

- title
- content
- actions
  - Widget listesi
  - Genellikle Button / TextButton



Ekranda bazı uyarıların verilmesi istendiğinde AlertDialog kullanılabilir. Diyaloğun başlığı içeriği ve aksiyonları belirtilebilir. Genellikle Actions kısmına tuşlar eklenip bu tuşların seçimine göre bir işlem gerçekleştirilir.

## Padding

- Herhangi bir widget içerisinde bir gruba padding vermek
- Padding desteği sunmayan arayüz elementleri için
- padding: -> EdgeInsets
- child:



**Padding**  
A widget that insets its child by the given padding.

Container gibi bazı bileşenlerde padding ve margin verilebileceği gibi yalnızca bir widget için padding de verilebilmektedir. Widget'ın kendi padding desteği bulunmuyorsa o widget'a Child olarak verilen alt widget padding ile sarmalanarak kullanılabilir.

## SizedBox

- Belirli widget'lar için boyut sınırı olması gerekir
  - Örn TextField
- Boyut sınırı olmayan bir container içerisinde boyut vermek
- child
- height
- width

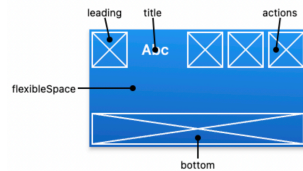


**SizedBox**  
A box with a specified size. If given a child, this widget forces its child to have a specific width and/or height (assuming values are...)

Uygulamada boyutu sabit olması gereken bileşenler bulunuyorsa height ve width gibi özellikler ile alt bileşenin boyutunun sabitlenmesini sağlar.

## AppBar

- leading
- title
  - Widget
- titleSpacing
- actions
  - Widget Listesi



Uygulamalar çalıştırıldığında uygulamanın logosunun, adının, gerçekleştirdiği belirli işlemlerin ve bilgiyi gösterdiği alanların bulunduğu bir AppBar tanımlanabilir. Uygulamanın anasayfasında her zaman gözüken bir menü işlevi de görebilmektedir.

## BottomSheet

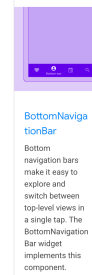
- Persistent
  - ScaffoldState.showBottomSheet()
  - Her zaman gözükür
- Modal
  - showModalBottomSheet()
  - Gerektiğinde bir menü yerini alır



Uygulamaların altında kalıcı olarak seçilebilecek bir menü olarak veya istendiğinde gösterilebilecek bir pencere olarak BottomSheet kullanılabilir. Persistent yapıldığında her zaman gözükürken Modal yapıldığında showModalBottomSheet() metodu tetiklendiğinde gösterilir.

## BottomNavigationBar

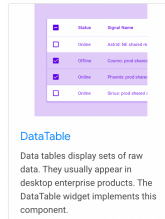
- View'lar arası geçiş
  - Genellikle 3 ile 5 arasında
- Özellikler
  - currentIndex -> int
  - elevation -> double
  - iconSize -> double
  - items -> BottomNavigationBarItem(icon, label)
- Olay
  - onTap



Uygulamada sayfalar arasında geçişte kullanılan ve her zaman gözükten bir bardır. Genellikle ideal olarak 3 ile 5 arasında olması istenir. Uygulama çalıştırıldığında varsayılan olarak seçili index'in veya o anki index'in elde edilmesi ve belirlenmesi mümkündür. Elemanları items özelliği ile verilir ve her bir item BottomNavigationBarItem olmalıdır.

## DataTable

- columns
  - DataColumn listesi
- rows
  - DataRow listesi
- DataRow
  - cells
    - DataCell listesi
- DataCell
  - selected
  - onSelectChanged



Veri ağırlıklı uygulamalarda sütun verilerinin listelenmesi, seçilebilmesi, ekleme silme ve güncellemede takip edilebilmesi adına DataTable kullanılabilir. Herhangi bir satırın, sütunun veya hücrenin seçimi gerçekleştirilebilmektedir.

## Cupertino

- iOS stili widget'lar
- Belirli özelliklerin deęiřmesi
- Temalar için MaterialApp yerine CupertinoApp
- İsimler benzer

Material stilinde kullanılan widget'ların yerine Cupertino widget'lar kullanılabilir. Oldukça benzer isimler ile aynı widget'ların Cupertino versiyonları bulunmaktadır. Bu sayede kodda büyük bir deęişiklik gerçekleştirilmeden iOS stili uygulamalar geliřtirmek de mümkündür.