

# 2. Ünite: Ekran Tasarımı



## Konu: TextView (Metin Görünümü)

Merhaba arkadaşlar!

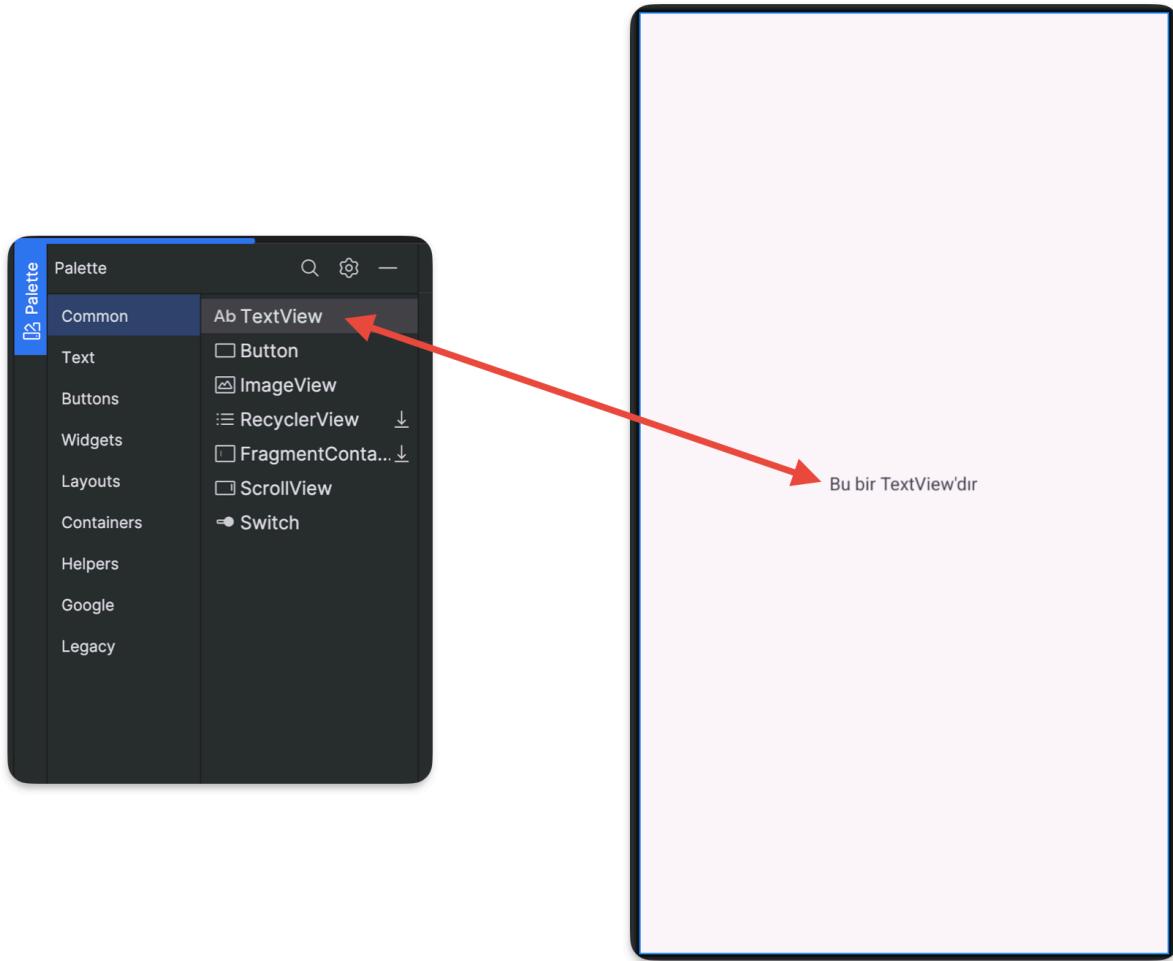
Bugünkü dersimizde Android uygulamalarının en çok konuşan, en çok bilgi veren elemanını inceleyeceğiz: TextView. Uygulamanızda kullanıcıya bir şeyler söylemek istiyorsanız (örneğin: "Hoş geldiniz", "Skor: 100" veya "Lütfen bekleyiniz"), kullanacağınız araç budur.

### 1. TextView Nedir?

TextView, mobil cihazın ekranında metin göstermek için kullanılan en temel bileşendir<sup>1</sup>.

- **Kullanıcı değiştiremez:** Sadece okuma amaçlıdır (Read-only).
- **Her yerde kullanılabilir:** Başlıklarda, açıklamalarda, etiketlerde...

|| Benzetme: Müzedeki tabloların altındaki bilgi kartları gibidir. Bakarsınız, okursunuz ama üzerini karalayamazsınız.



## 1.1. Tasarım Ekranına Ekleme Yöntemleri

Bir TextView'ı ekrana iki şekilde koyabiliriz:

### A. Sürükle-Bırak Yöntemi (Palette)

En kolayı budur.

1. Android Studio'da **Design** sekmesine geçin.
2. Sol taraftaki **Palette** panelini açın.
3. **Common** veya **Text** kategorisine tıklayın.
4. **TextView** ögesini tutup telefon ekranına sürükleyin.

### B. XML Kodlama Yöntemi

Gerçek bir yazılımcı gibi kod yazarak eklemek istersek `activity_main.xml` dosyasının **Code** kısmına şunu yazarız:

XML

```
<TextView  
    android:id="@+id/benimMetnim"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Merhaba Gençler!" />
```

## 1.2. TextView Özellikleri (Attributes)

Bir TextView'ın rengini, boyutunu ve stilini değiştirmek için "Niteliklerini" (Attributes) kullanırız. İşte en sık kullanacağınız4:

Özellik (Attribute)	Ne İşe Yarar?	Örnek Değer
android:id	Java kodunda bu elemanı bulmak için kullandığımız kimlik numarasıdır.	@+id/baslikText
android:text	Ekranda kullanıcıya gösterilecek yazıdır.	"Bilişim Teknolojileri"
android:textSize	Yazının büyüklüğüdür. Genelde sp birimi kullanılır.	"24sp"
android:textColor	Yazının rengidir. Renk kodları (Hex) kullanılır.	"#FF0000" (Kırmızı)
android:textStyle	Yazıyı kalın (bold) veya eğik (italic) yapar.	"bold"
android:textAllCaps	true yapılrsa, küçük harfle yazsanız bile hepsi BÜYÜK HARF görünür.	"true"
android:padding	Metnin kendi kutusu içindeki kenar boşluğudur (İç boşluk).	"20dp"

## 1.3. Örnek Uygulama: Renkli ve Büyük Yazı

Hadi kodla oynayalım! XML tarafında bir TextView'ı şöyle süsleyebiliriz. Bu kodu Split veya Code ekranına yapıştırıp sonucu görebilirsiniz:

XML

```
<TextView  
    android:id="@+id/ozelMetin"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Bilişim"
```

```
    android:textSize="50sp"  
    android:padding="25dp"  
    android:textColor="#0000EE" />
```

Bu kod ekranda mavi renkli, 50sp boyutunda dev bir "Bilişim" yazısı oluşturur.

## 1.4. Java ile Müdahale Etmek (Programatik Yaklaşım)

Tasarım ekranında yazdığımız yazı hep sabit kalır. Ama biz program çalışırken (örneğin bir butona basınca) yazının değişmesini istersek **Java** kodlarını kullanırız12.

Bunun için `MainActivity.java` dosyasına gideriz:

Adım 1: Tanımlama ve Bağlama

Once Java'ya "Benim tasarımımında bir TextView var, onu bul" demeliyiz.

Java

```
// 1. TextView sınıfından bir değişken oluştur  
TextView benimYazim;  
  
// 2. onCreate metodunun içinde ID'si ile bul ve eşleştir  
benimYazim = findViewById(R.id.ozelMetin);
```

Adım 2: İçeriği Değiştirme

Artık `benimYazim` değişkeni ile ekrandaki yazıya hükm edebiliriz13.

Java

```
// Metni "Kodlamayı Seviyorum" olarak değiştirir.  
benimYazim.setText("Kodlamayı Seviyorum");
```

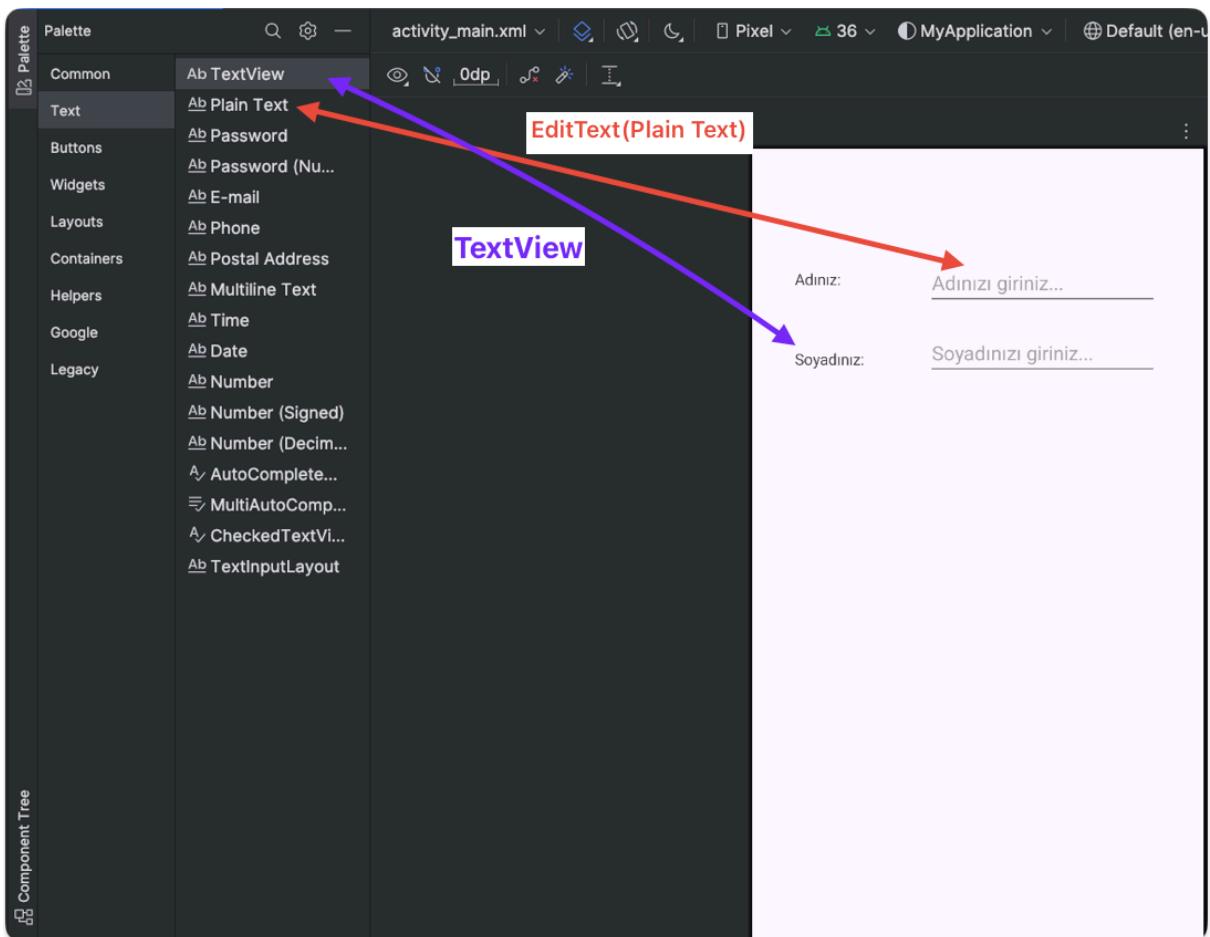
## 2. EditText Nedir?

`TextView` ile ekranı yazı yazmıştık ama kullanıcı buna cevap veremiyordu. Şimdi `EditText` ile kullanıcıdan veri alıcaz! Bir üye kayıt formu, bir mesajlaşma kutusu veya hesap makinesi yapmak istiyorsanız, `EditText` sizin en iyi dostunuzdur.

`EditText`, kullanıcının içine metin veya sayı girebildiği, düzenlenebilir bir metin kutusudur. Aslında o da bir `TextView`'dır ama "yazılabilir" özellikler eklenmiştir.

- Klavye Kontrolü:** Tıklandığında otomatik olarak klavye açılır.
- İpucu (Hint):** Kutucuk boşken kullanıcıya ne yazması gerektiğini silek bir yazıyla söyler.

Benzetme: Sınav kağıtlarındaki "Adınız Soyadınız: ....." boşluğu gibidir. Oraya siz yazarsınız.



## 2.1. En Önemli Özellikler (Attributes)

EditText'i EditText yapan özellikler şunlardır:

Özellik (Attribute)	Ne İşe Yarar?	Örnek
<code>android:id</code>	Java'da bu kutudaki yazıyı almak için kullanacağımız kimlik.	<code>@+id/editKullaniciAdi</code>
<code>android:hint</code>	Kutu boşken görünen silik ipucu yazısı.	<code>"Adınızı giriniz"</code>

`android:inputType`

**(Çok Önemli!)** Klavyenin türünü ve yazının görünüşünü belirler.

`text` , `number` ,  
`textPassword`

## 2.2. InputType: Hangi Veriyi Alacağız? 🔑

İste burası çok kritik! Kullanıcıdan ne istediginize göre `inputType` özelliğini değiştirmeniz gereklidir. Bu özellik, telefonda açılacak klavyeyi de değiştirir.

### A. Düz Metin ( `textPersonName` veya `text` )

En temel halidir. Harf, sayı, sembol... Her şeyi yazabilirsiniz.

- **Kullanım:** İsim, Mesaj, Adres.
- **Klavye:** Standart harf klavyesi açılır.

### B. Sadece Sayı ( `number` )

Kullanıcının sadece rakam girmesini sağlar. Harf girmeye çalışsa bile klavyede harfler çıkmaz.

- **Kullanım:** Yaşı, Telefon Numarası, Miktar.
- **Klavye:** Sadece rakamların olduğu klavye açılır.

### C. Şifre ( `textPassword` )

Kullanıcının yazdığı karakterleri gizler (nokta veya yıldız şeklinde gösterir).

- **Kullanım:** Üye girişi, Pin kodu.
- **Görünüm:** Yazdıklarınız ..... şeklinde görünür.

## 2.3. Kodlayalım! (XML ve Java)

### Tasarım (XML) Kısmı

Hadi bir giriş ekranı için kullanıcı adı ve şifre kutusu yapalım.

XML

```
<EditText  
    android:id="@+id/editAd"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"
```

```
    android:hint="Kullanıcı Adınızı Giriniz"
    android:inputType="textPersonName" />

<EditText
    android:id="@+id/editSifre"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Şifrenizi Giriniz"
    android:inputType="textPassword" />

<EditText
    android:id="@+id/editYas"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Yaşınız"
    android:inputType="number" />
```

## Kod (Java) Kısmı

Tasarımı yaptık, peki kullanıcının yazdığı yazıyı Java'da nasıl alacağız?

Java

```
// 1. Tanımla
EditText adKutusu;
EditText sifreKutusu;

// 2. Bağla (onCreate içinde)
adKutusu = findViewById(R.id.editAd);
sifreKutusu = findViewById(R.id.editSifre);

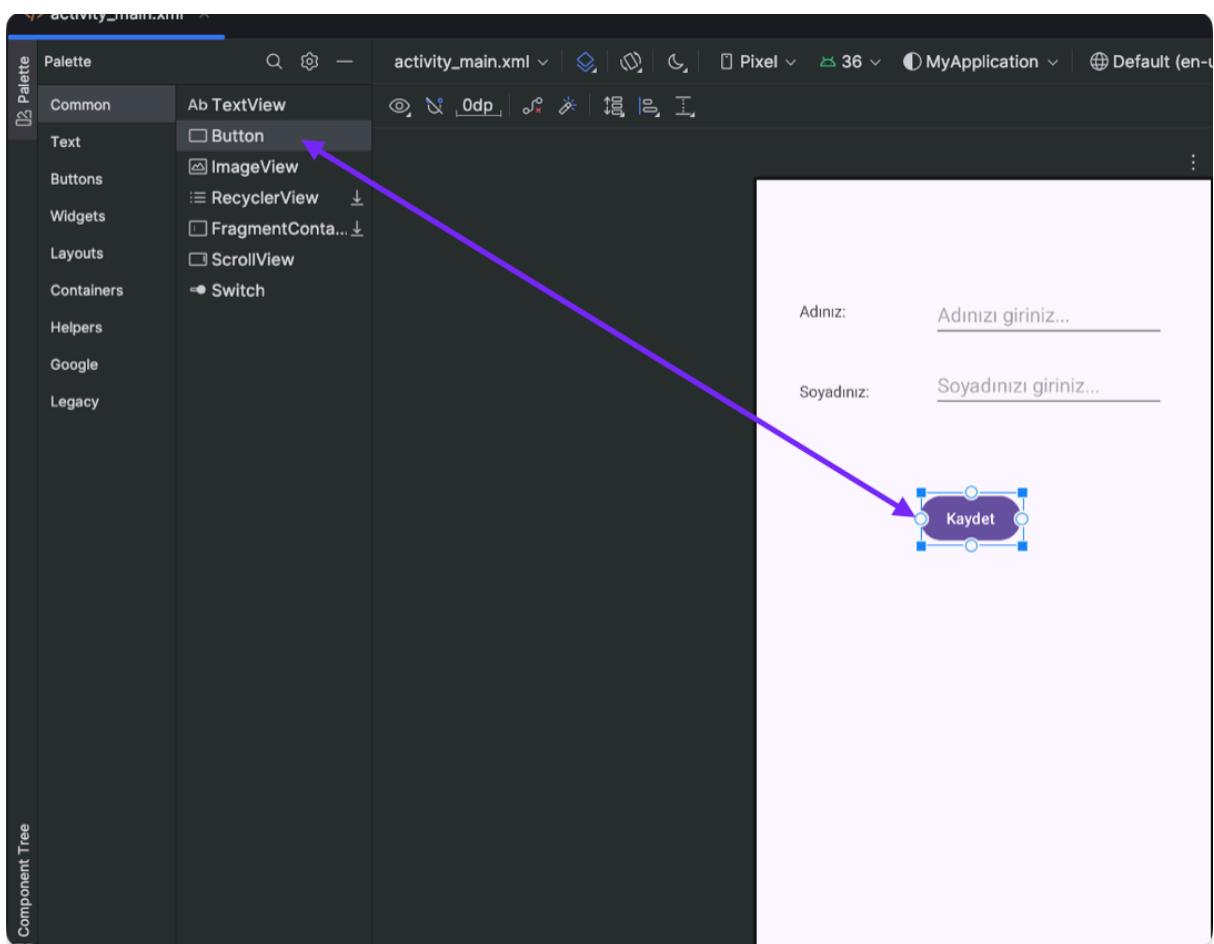
// 3. Veriyi Al (Genelde bir butona basılıncaya yapılır)
String gelenAd = adKutusu.getText().toString(); // "toString()" çok önemli!
String gelenSifre = sifreKutusu.getText().toString();
```

## 3. Button Nedir?

Button, üzerine tıklandığında veya dokunulduğunda bir **eylemi (işlemi)** başlatan bileşendir.

- **Tetikleyicidir:** Bir kod bloğunu çalıştırmak için "Başla" işaretini verir.
- **İnteraktiftir:** Tıklandığında renk değiştirebilir veya ses çıkarabilir (Geri bildirim verir).

Günlük Hayat Benzetmesi: Evinizdeki lamba anahtarı gibidir. Anahtara (Button) basarsınız ve elektrik (Kodlar) devreye girip lambayı (Eylemi) yakar.



### 3.1. Tasarım Ekranına Ekleme

#### A. Sürükle-Bırak (Palette)

1. **Palette** panelinden **Buttons** kategorisine gelin.
2. En üstteki **Button** ögesini tutup ekranın istediğiniz yerine bırakın.
3. *Unutmayın:* ConstraintLayout kullanıyorsanız, butonun sağ-sol-üst-alt yaylarını bağlamayı unutmayın!

#### B. Kod ile Ekleme (XML)

Tasarımcı gibi düşünmek istiyorsanız kod tarafında şöyle yazarız:

XML

```
<Button  
    android:id="@+id/btnGonder"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Mesajı Gönder"  
    android:backgroundTint="#FF5722" />
```

### 3.2. Button'ın Süper Güçleri (Attributes)

Button da bir `View` olduğu için `layout_width` veya `layout_height` gibi özellikler vardır. Ama onu özel yapan bazı nitelikler şunlardır:

Özellik (Attribute)	Ne İşe Yarar?	Örnek
<code>android:text</code>	Düğmenin üzerinde yazacak yazı.	"Giriş Yap"
<code>android:backgroundTint</code>	Düğmenin arka plan rengi.	"#4CAF50" (Yeşil)
<code>android:onClick</code>	<b>(Çok Önemli!)</b> Düğmeye tıklandığında Java'da hangi fonksiyonun çalışacağını belirtir.	"tiklamalslemi"
<code>android:enabled</code>	Düğmenin çalışıp çalışmayacağını belirler. <code>false</code> yaparsanız sönük görünür ve tıklanamaz.	"true" / "false"

### 3.3. Tıklama Olayı

Bir butona tıklandığını anlamanın iki ana yolu vardır. Kitabımızda (Sayfa 64) en pratik yöntem olan **XML üzerinden bağlama** anlatılmıştır.

#### Yöntem 1: XML `onClick` Özelliği (Eski Yol)

Adım 1: XML Dosyasına Emir Verin

Butonun koduna `android:onClick="selamVer"` satırını ekleyin. Bu, "Bana tıklanırsa Java'daki selamVer isimli метод bul ve çalıştır" demektir.

XML

```
<Button  
    android:id="@+id/btnSelam"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Selamla"  
    android:onClick="selamVer" /> ...
```

\*\*Adım 2: Java'da Karşılıyın\*\*

`MainActivity.java` dosyasına gidin ve aynı isimle bir metot (fonksiyon) oluşturun.

## Java

```
// Bu metodun ismi XML'deki onClick ile AYNI olmalıdır!  
public void selamVer(View view) {  
    // Tıklanınca ne olsun?  
    Toast.makeText(this, "Merhaba Dünya!", Toast.LENGTH_SHORT).show();  
}
```

Dikkat: Metodun public olması, void olması ve parantez içinde (View view) parametresini alması zorunludur. Yoksa uygulama çöker! 💥

## Yöntem 2: Java Listener (Yeni Yol)

Bu yöntem daha çok kod yazmayı gerektirir ama daha esnektir (İleride çok kullanacağınız).

## Java

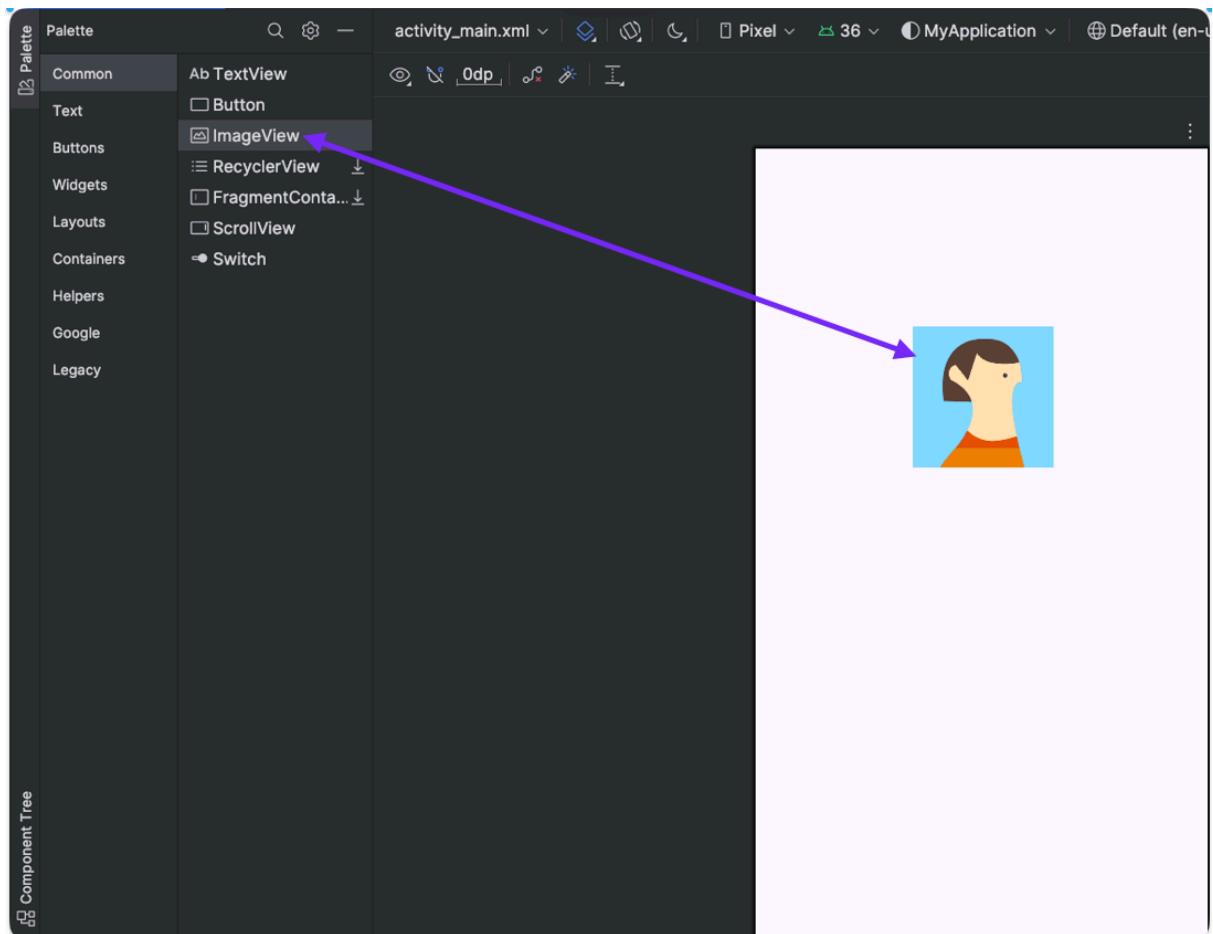
```
// 1. Butonu Tanımla  
Button benimButonum = findViewById(R.id.btnSelam);  
  
// 2. Dinleyici (Listener) Tak  
benimButonum.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        // Buraya tıklama kodları gelecek  
        Toast.makeText(getApplicationContext(), "Butona Tıklandı!", Toast.LEN
```

```
GTH_LONG).show();  
}  
});
```

## 4. ImageView Nedir?

Uygulama arayüzünde resim dosyalarını (PNG, JPG, GIF vb.) görüntülemek için kullanılan bileşendir1.

Benzetme: Evinizdeki boş fotoğraf çerçevesi gibidir. İçine istediğiniz fotoğrafı koyabilirsiniz, değiştirebilirsiniz ama çerçeve (ImageView) orada durur.



### 4.1. İlk Adım: Resmi Projeye Ekleme

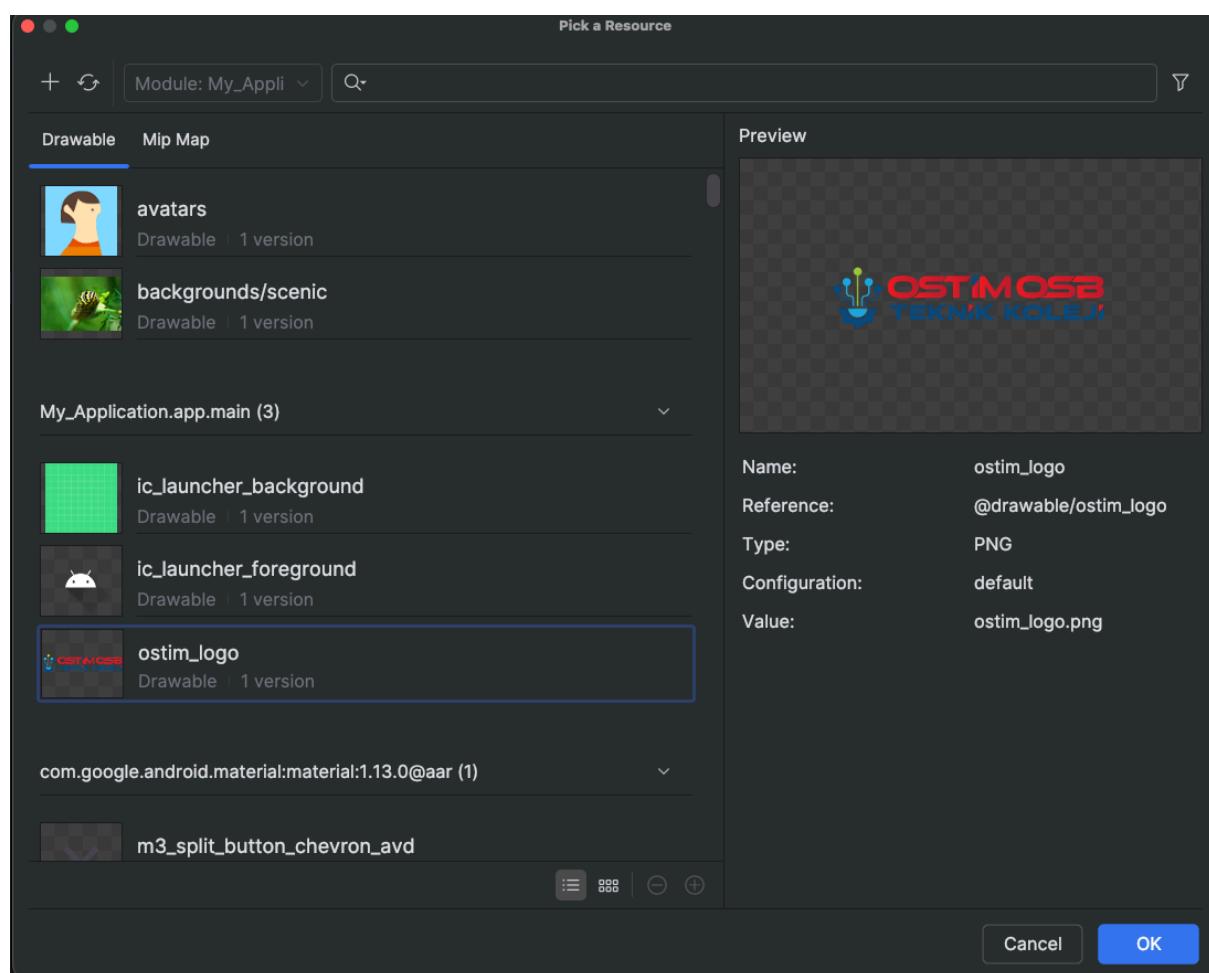
Kod yazmadan önce, kullanacağımız resmi Android Studio'nun içine atmamız gereklidir. Bu işlem çok kritiktir!

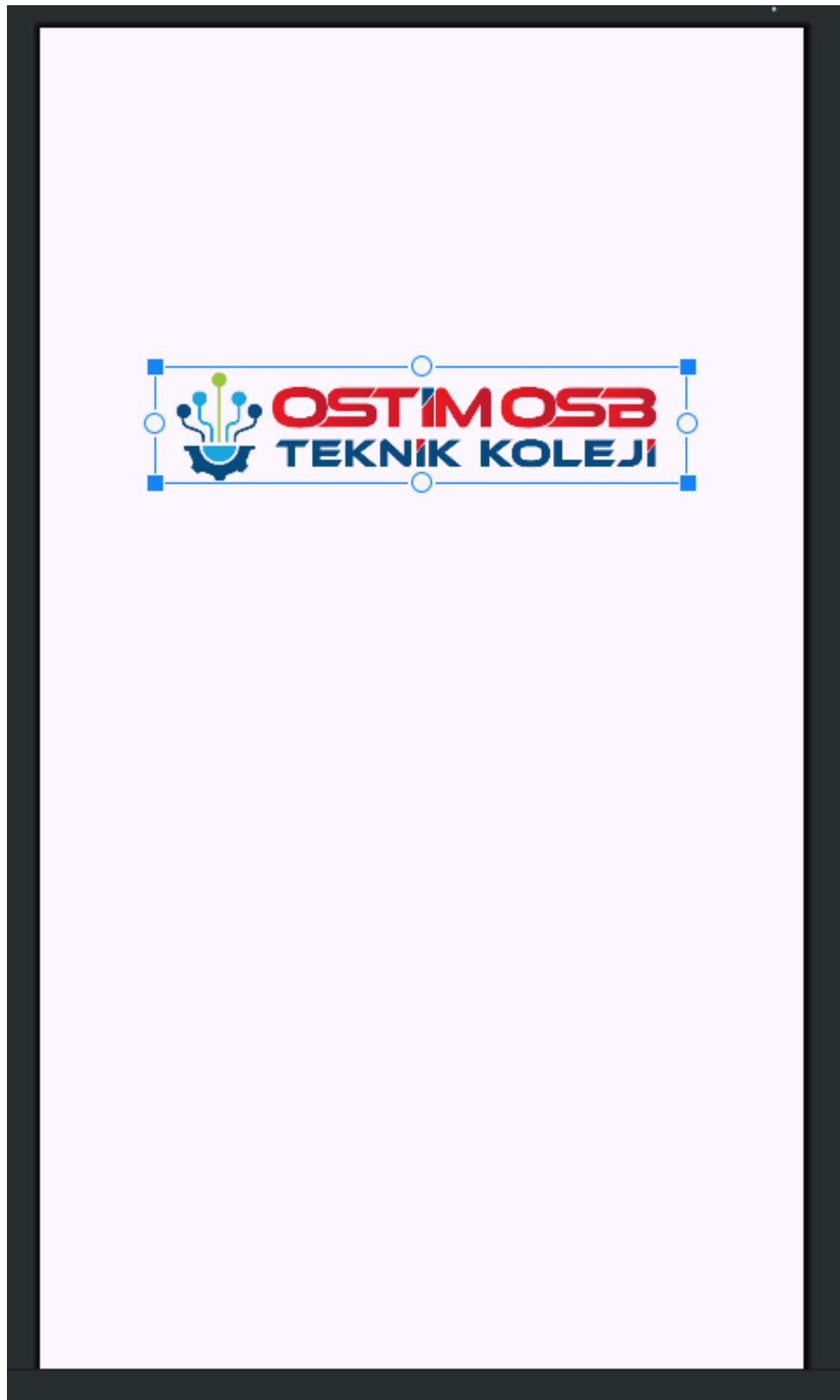
#### Nasıl Yapılır?

1. Bilgisayarınızdaki resmi kopyalayın (Ctrl+C).
2. Android Studio'da sol taraftaki proje ağaçından **res** klasörünü açın.
3. **drawable** klasörüne sağ tıklayın ve **Paste (Yapıştır)** deyin.
4. Çıkan kutucukta **drawable** seçeneğini seçin (drawable-v24 değil!).

**⚠️ ÇOK ÖNEMLİ KURALLAR:**

- Resim isimlerinde **Türkçe karakter** (ç, ş, ī, ö, ü, ğ) kullanmayın!
- **Boşluk** bırakmayın (yerine **\_** alt çizgi kullanın).
- **İsim sayı** ile başlamamalı.
- *Doğru Örnek:* **okul\_manzarasi.jpg**
- *Yanlış Örnek:* **Okul Manzarası 1.jpg**





---

## 4.2. Tasarım Ekranına Ekleme

### A. Palette Yöntemi

1. **Palette** panelinden **Common** veya **Widgets** sekmesine gelin.
2. **ImageView** ögesini tutup ekrana sürükleyn.
3. Bıraktığınız anda size "Hangi resmi koyayım?" diye soran bir pencere açılır ( **Pick a Resource** ). Buradan **drawable** 'a attığınız resmi veya hazır ikonları seçebilirsiniz.

## B. XML Kodlama Yöntemi

`activity_main.xml` dosyasının **Code** kısmına şunu yazarak da ekleyebilirsiniz:

XML

```
<ImageView  
    android:id="@+id/resimKutusu"  
    android:layout_width="match_parent"  
    android:layout_height="200dp"  
    android:src="@drawable/meblogo" />
```

Not: `src` (source), kaynağı belirtir. `@drawable/resim_adi` diyerek resmimizi çağrıriz3.

### 4.3. ImageView'ın Süper Güçleri (Attributes)

ImageView'ı yönetmek için şu özellikleri çok sık kullanırız4:

Özellik (Attribute)	Ne İşe Yarar?
<code>android:src</code>	Ekranda gösterilecek resmin kaynağını/adresini belirtir.
<code>android:scaleType</code>	(En Önemli!) Resmin çerçevesinin içine nasıl sığacağını belirler.
<code>android:maxWidth</code>	Resmin olabileceği en büyük genişliği sınırlar.
<code>android:maxHeight</code>	Resmin olabileceği en büyük yüksekliği sınırlar.
<code>android:tint</code>	Resmin üzerine renk filtresi atar (Örneğin siyah bir ikonu kırmızıya boyamak için).

### 4.4. scaleType: Resim Nasıl Sığacak?

Bazen resim çerçeveden büyük, bazen küçük olabilir. `scaleType` özelliği ile resmin nasıl davranışacağını seçeriz. İşte en popüler seçenekler5:

- `fitXY` : Resmi çerçevenin boyuna uydurmak için **sündürür (gerer)**. Resim basık veya uzun görünebilir. (En-boy oranı bozulur).

- **centerCrop** : Resmi, çerçeveyi tamamen dolduracak şekilde büyütür ve ortalar. Fazlalık kısımları **kırpar (keser)**.
- **fitCenter** : Resmin tamamını gösterir ve ortalar. Resim bozulmaz ama kenarlarda **boşluk kalabilir**.
- **fitStart / fitEnd** : Resmi bozmadan çerçevenin başına (sol/üst) veya sonuna (sağ/alt) yaslar.

## 4.5. Java ile Resim Değiştirme

Uygulama çalışırken bir butona basıldığında resmi değiştirmek istersek ne yapacağız?

**Senaryo:** "Değiştir" butonuna basınca manzara resmi yerine logo gelsin.

Java

```
// 1. Tanımla
ImageView benimResmim;
Button butonDegistir;

// 2. Bağla (onCreate içinde)
benimResmim = findViewById(R.id.resimKutusu);
butonDegistir = findViewById(R.id.btnDegistir);

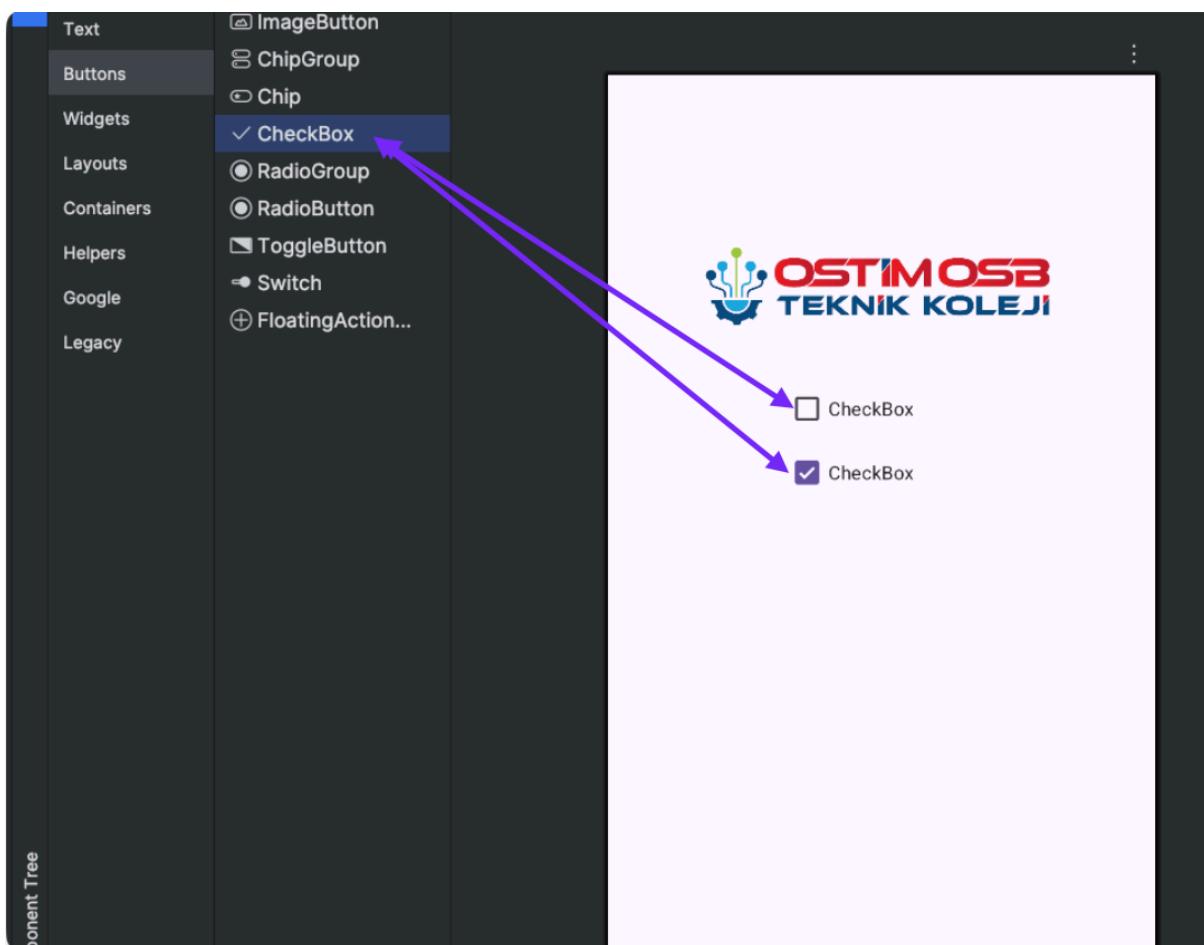
// 3. Değiştir (Butona tıklanınca)
butonDegistir.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // "yeni_resim" ismindeki dosyayı göster
        benimResmim.setImageResource(R.drawable.yeni_resim);
    }
});
```

## 5. CheckBox Nedir?

CheckBox, kullanıcının bir veya birden fazla seçeneği işaretlemesine (veya işaretini kaldırmasına) olanak tanıyan bir bileşendir.

- **Çoklu Seçim:** `RadioButton`'dan farklı budur; birden fazla CheckBox aynı anda seçilebilir.
- **İki Durumlu:** Sadece iki hali vardır: Seçili (Checked) veya Seçili Değil (Unchecked).

Günlük Hayat Benzetmesi: Bir pizza siparişi verirken "Mantar olsun", "Sucuk olsun", "Biber olsun" diye seçtiğiniz o liste var ya? İşte onlar CheckBox'tır! 🍕



## 5.1. Tasarım Ekranına Ekleme

### A. Palette Yöntemi

1. **Palette** panelinden **Buttons** kategorisine gelin.
2. **CheckBox** ögesini tutup ekrana sürükleynin.
3. İçindeki yazılımı değiştirmek için **Attributes** panelinden `text` özelliğini bulun.

### B. Kod ile Ekleme (XML)

`activity_main.xml` dosyasının **Code** kısmına şunu yazarak da ekleyebilirsiniz:

XML

```
<CheckBox  
    android:id="@+id/checkMuzik"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Müzik Dinlemek"  
    android:checked="false" />
```

## 5.2. CheckBox'ın Süper Güçleri (Attributes)

CheckBox'ı yönetmek için şu özellikleri kullanınız:

Özellik (Attribute)	Ne İşe Yarar?	Örnek
<code>android:text</code>	Kutucuğun yanında yazacak açıklama metni.	"Kitap Okumak"
<code>android:checked</code>	Başlangıçta kutucuğun işaretli gelip gelmeyeceğini belirler.	"true" (İşaretli) / "false" (Boş)
<code>android:buttonTint</code>	Kutucuğun rengini değiştirir.	"#FF0000" (Kırmızı kutucuk)

## 5.3. Java ile Kontrol Etme: "Seçtin mi, Seçmedin mi?" 🧑

Kullanıcının kutuyu işaretleyip işaretlemediğini anlamak için Java'da `isChecked()` komutunu kullanırız. Bu komut bize `true` (seçili) veya `false` (seçili değil) cevabını verir.

**Senaryo:** Bir "Hobiler" uygulaması yapalım. Kullanıcı "Spor" kutusunu seçip butona basınca ekrana mesaj verelim.

**XML Tarafı:**

XML

```
<CheckBox  
    android:id="@+id/checkSpor"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"
```

```
    android:text="Spor Yapmak" />

<Button
    android:id="@+id/btnKontrol"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Kaydet" />
```

### Java Tarafı (`MainActivity.java`):

Java

```
// 1. Tanımla
CheckBox sporKutusu;
Button kaydetButonu;

// 2. Bağla (onCreate içinde)
sporKutusu = findViewById(R.id.checkSpor);
kaydetButonu = findViewById(R.id.btnKontrol);

// 3. Kontrol Et (Butona tıklanınca)
kaydetButonu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Kutucuk işaretli mi?
        if (sporKutusu.isChecked()) {
            Toast.makeText(getApplicationContext(), "Sporu seviyorsun, harika!", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "Belki spora başlamalısın.", Toast.LENGTH_SHORT).show();
        }
    }
});
```