

# Android Adlandırma Kuralları

Android dili için, Google tarafından henüz belirli bir adlandırma kuralları **belirtilmemiştir**. Bundan dolayı bu makalede geliştiriciler tarafından **yaygın biçimde kullanılan** adlandırma kurallarından bahsedeceğiz.

## Google Tarafından Önerilen Adlandırma Kuralları

Google **sadece kod kısmı için** adlandırma kurallarını önermiştir. Bu kurallar aşağıdaki gibidir:

1. **non-public, non-static** field isimleri **m** harfi ile başlar.
2. **static** field isimleri **s** harfi ile başlar.
3. **Diğer** field isimleri **küçük harfle** başlar.

**public static final** fields (sabit değişkenler) tüm harfleri şu şekilde yazılır: **ALL\_CAPS\_WITH\_UNDERSCORES**.

**Örnek:**

```
1 public class MyClass {
2     public static final int SOME_CONSTANT = 42;
3     public int publicField;
4     private static MyClass sSingleton;
5     int mPackagePrivate;
6     private int mPrivate;
7     protected int mProtected;
8 }
```

## Layout

Activity sınıfları için: act\_activity\_adi

**Örnek:** MainActivity isminde bir Activity sınıfı varsa, bu sınıfın layout ismi: **act\_main**

View sınıfları için: view\_id

**Örnek:** TextView türünden txtMain isminde bir nesne yaratmışsak, kullanılacak layout dosya adı **txt\_main** olabilir.

Dialog sınıfı için: dlg\_id

**Örnek:** Dialog sınıfı türünden dlgMain isminde bir nesne yaratmışsak, kullanılacak layout dosya adı **dlg\_main** olabilir.

## String IDs

<string name="act\_about\_us\_dlg\_help1\_title"></string> formatına göre isimlendirme yapılır. Örneğin Activity sınıfımız AboutUsActivity ismine sahip olursa, bu sınıfta kullanılan bir Dialog nesnesinin(değişken adı dlgHelp ise) başlığına string.xml dosyasında şu şekilde referans verebiliriz:

**act\_about\_us\_dlg\_help\_title**

**Not:** Eğer bir string değeri tüm uygulama boyunca kullanılacaksa, örneğin btnTamam şeklinde kullanılacak bir buton nesnesini string.xml dosyasında tanımlarken, **global\_btn\_tamam** şeklinde id verebiliriz.

## IDs

Xml dosyalarında viewlerin id'leri genelde şu şekilde kullanılır: **@id/list\_apple**. Bu değişkeni kod kısmında kullanmak istediğimiz zaman şu şekilde erişebiliriz:

```
1 | ListView listApple= (ListView)findViewById(R.id.list_apple)
```

## Drawable

drawable klasörlerine eklenen bir ikonu, **clickable(tıklanabilir)** özellikli kullanacaksa, bu ikonu adlandırırken **ic\_action\_delete** gibi kullanmak mantıklı olacaktır. Yani action özelliğine sahip ikonlar **ic\_action** ön eki ile başlayabilir.

Örneğin **action bar** kısmında kullanılacak bir resmin adı **ab\_bottom\_solid\_example.9.png** şeklinde olur. Burada **ab** ön eki ile başlaması **action bar** kelimelerin ilk harflerini ifade eder.

**Tema** için kullanılan resimler ise **apptHEME** ön eki ile başlar. **Örnek:** `apptHEME_btn_default_focused_holo_light.9.png`

**Not:** Uygulamanın ikon adı **ic\_launcher.png** şeklinde adlandırılır.

## Sonuç

Android uygulamalar geliştirirken yukarıdaki adlandırma kurallarına uyarsak, belirli bir standart içerisinde kod yazmış oluruz. Bu sayede diğer geliştiricilerin yazmış olduğu kodları da rahat bir şekilde anlayabiliriz.