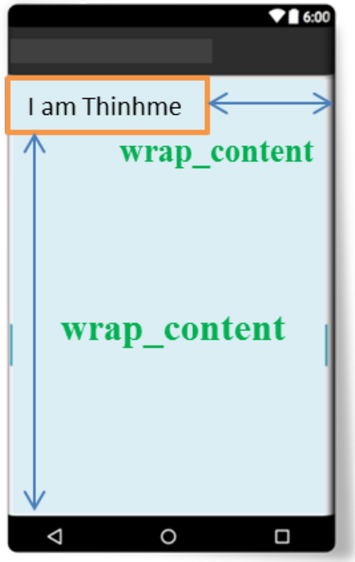
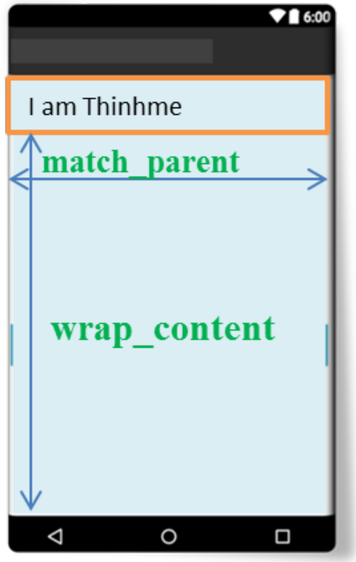
LAYOUT OLUŞTURMA

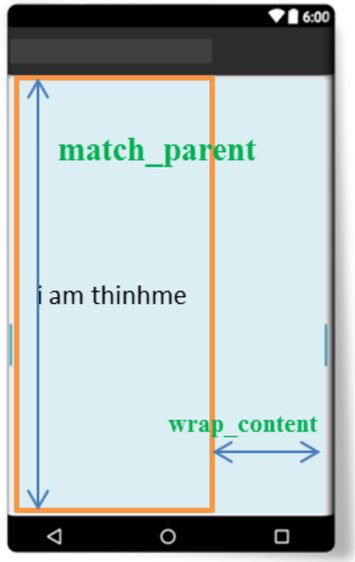
- Layout Android uygulama ekranlarındaki görsel sayfalardır.
- Sayfadaki eklediğimiz öğelerin kaplayacağı width ve height özellikleri “wrap_content” veya “match_parent” olarak ayarlanabilir. Bu seçilen özelliklere göre farklı boyutlardaki cihazlarda öğenin kaplayacağı alan baştan belirlenir.



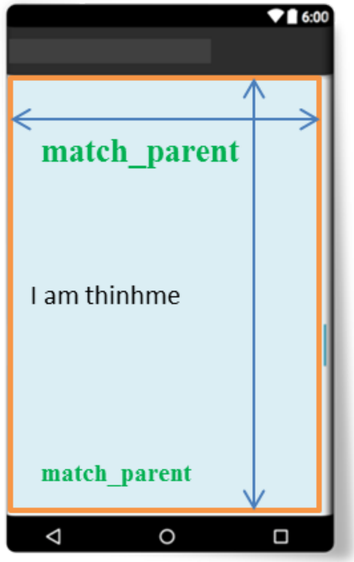
android:layout_width = “wrap_content”
android:layout_height = “wrap_content”



android:layout_width = “match_parent”
android:layout_height = “wrap_content”



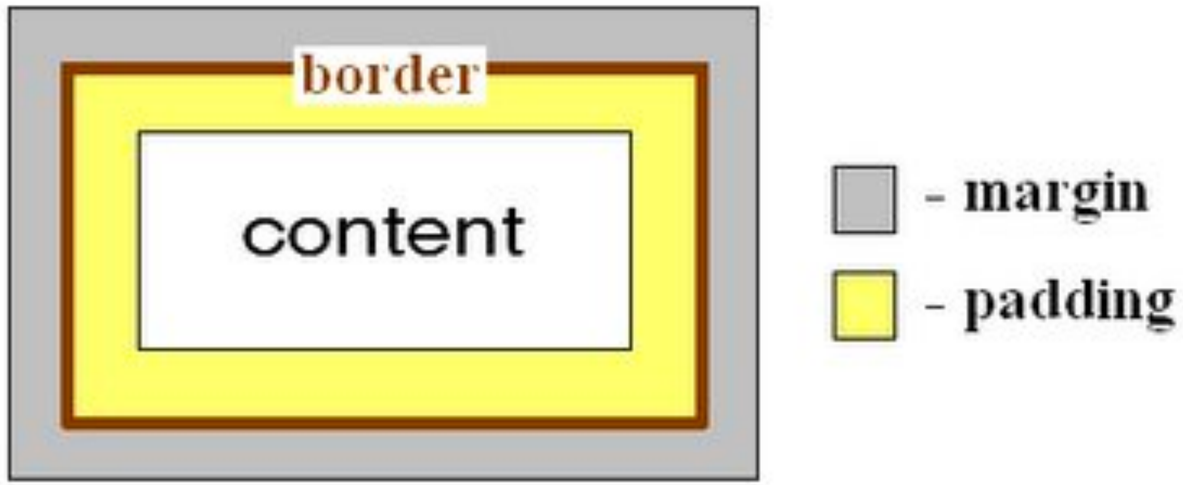
android:layout_width = “wrap_content”
android:layout_height = “match_parent”



android:layout_width = “match_parent”
android:layout_height = “match_parent”

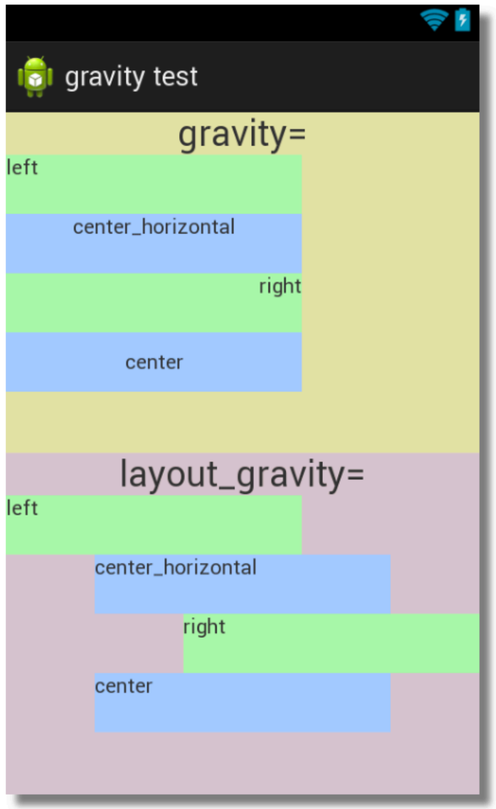
Padding ve Margin

- Padding öğenin iç boşluğunu, layout_margin öğenin dış çerçeveden boşluğunu temsil eder.



-

- Eklenen öğenin içeriğinin konumunu gravity özellikleri ile değiştirebiliriz. Layout_gravity ile öğeyi ekranda konumlandırabiliriz.



Linear Layout

- Linear Layout tasarım prensipleriyle çalışmak isterseniz layout klasörüne sağ tıklayıp yeni bir layout resource file'dan linear layout xml dosyası oluşturabilirsiniz.
- Eklenen tüm öğeler yan yana eklenecektir. Bu özelliği alt alta çevirmek için component tree'den linearLayoutCompat'ın orientation özelliğini değiştirmek gerekir.
- Öğelerin layout_width veya layout_height özellikleri değiştirilebilir.
- LinearLayoutCompat'ın gravity özelliğini değiştirerek öğeler istenilen konuma hizalanabilir.
- Xml code kısmında ana ekranın android:weightSum="100" değeri belirlenerek diğer öğelerin android:weightSum değeri belirlenerek ağırlığı ayarlanabilir.

Constraint Layout

- En çok tercih edilen ve sürekli güncellenen bir layout'dur. Varsayılan olarak ilk açılan activity_main layout'u constraint layout'dur.
- Herhangi bir öğeyi ekledikten sonra sağ tıklayıp "center" özelliğinden ortalama işlemleri yapılabilir.
- Arka plan rengi için values klasörü içindeki colors.xml dosyasına istenilen renk eklenebilir. Daha sonra bu sayfa yapısındaki background alanından seçilerek ayarlanabilir.
- TextBox Android de Plain Text olarak eklenebilmektedir.
- Tüm öğelerin background özelliğinde ve textcolor özelliğinden renkler değiştirilebilir.

- Eklenen her öge yatay ve dikeyde bir noktaya sabitlenmelidir. Sabitlenmediğinde yanında bir kırmızı uyarı çıkar ve proje çalıştığında kendini sayfanın en üst sağ noktasına konumlandırabilir.
- Sabitlemeleri öğenin etrafındaki yuvarlaklarda ayarlayıp istendiğinde koparabiliriz. Aynı şekilde properties'de layout penceresinden ayarlanabilir.

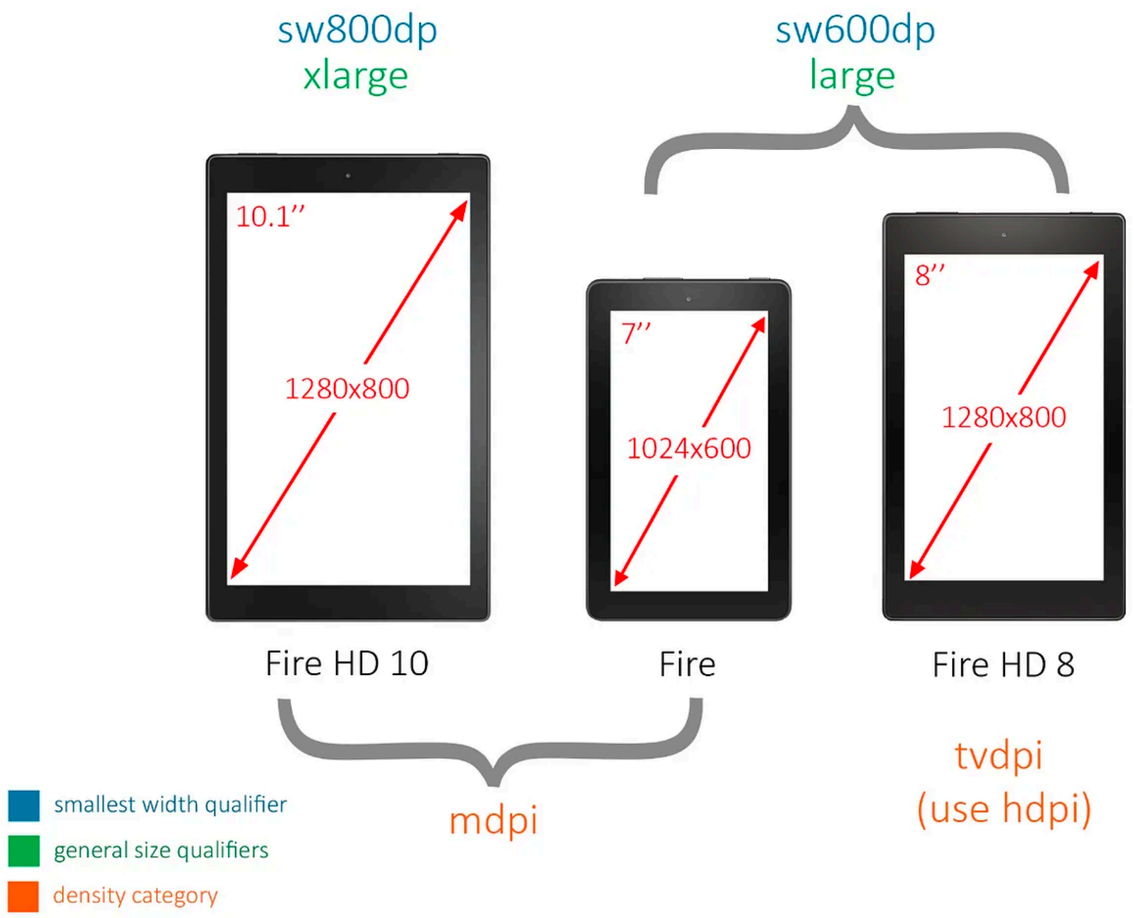
- Eklenen öğeler köşelerinden boyutlandırılabilir. Böylece layout_width ve layout_height değerleri değişir. “dp” değeri density pixcel anlamındadır. Ekranın boyutunun oranına göre kendini değiştirecek anlamına gelir. Bu alana “sp” (standart pixcel) değeri girilirse ekran değiştiğinde sabit boyutta kalır. Ekranın durumuna göre küçülüp büyümez. Yazılarda sp, görsel öğelerde dp tercih edilebilir.

- Görsel öğeler birbirine göre hizalanabilir. Hizalama işlemi sonucunda birbirine hizalanan öğelerin yeri değiştirilmek istenirse birlikte hareket ederler.
- Yönlenecek öğenin hangi yönde hizalanacak ise o yuvarlağı yönlendirilen öğeye bağlanır. Böylece yönlendirilen öge hareket edince diğeri de eder.
- Hizalamayı silmek için bağlantının yapıldığı yuvarlak seçilerek del tuşuna basılır ya da layout kısmından yuvarlağın olduğu yerdeki çarpıya basılır.
- İstenirse hizalanmak istenen öğeler seçilip sağ tıklanıp align sekmesinden istenilen hizalama yapılabilir.

- Sayfamıza kılavuz çizgileri eklenebilir. Böylece sayfanın istediğimiz alanına yerleşim yaparken bize yardımcı olur. Bunun için sayfaya sağ tıklayıp “add helpers” den yatayda ve dikeyde kılavuz çizgileri eklenebilir. Çizginin altındaki ok işaretinden gösterdiği değerler değiştirilebilir.

Çoklu Ekran Desteği

- Yaptığımız uygulama farklı ekran boyutları ve farklı çözünürlüklere göre tasarlanmalıdır.
- Constraint Layout ile mümkün olduğu kadar çoklu ekran desteği sağlanabiliyor.
- Ekranlar small, normal, large, xlarge olarak sınıflandırılabilir.
- Resimler için ise farklı ekran çözünürlük yoğunluklarına (dpi) göre bir sınıflandırma yapılılabilir.


sw800dp
xlarge
sw600dp
large
10.1''
1280x800
7''
1024x600
8''
1280x800
Fire HD 10
Fire
Fire HD 8
mdpi
tvdpi
(use hdpi)
smallest width qualifier
general size qualifiers
density category

- Çoklu ekran için project/app/src/main/res klasörü altında yeni android resource directory'den layout sınıfları oluşturularak her ekran boyutu için xml dosyası oluşturulur. Bu klasörler small, normal, large, xlarge olarak oluşturulur.
- Çoklu ekranlarda resimler için aynı şekilde drawable klasörleri eklenebilir. Bunlar midmap klasörlerindeki gibi xxxhdpi, xxhdpi, xhdpi, mdpi, hdpi olarak adlandırılır. Farklı çözünürlükte, aynı isimdeki resimler bu klasörlerin altına eklenerek projeye dahil edilebilir.

Çoklu Dil Desteği

- Telefonun bulunduğu bölgeye göre uygulama dilinin aynı olabilmesi için kullanılır. Hangi dile destek isteniyorsa o dilin ekini alan values dosyası oluşturulur (Örneğin values-en). Values dosyasında strings.xml içindeki öğelere isim verilerek dosya diğer values dosyalarına kopyalanır ve gerekli değişikliler yapılır. Daha sonra emülatörün “>>” işaretinin olduğu yerden dil için seçim yapılır.

Icon oluşturma

- Res dosyasında sağ tıklayıp image asset'ten yeni ikonlar oluşturulabilir. Yeni oluşturulan icon'lar mipmap dosyalarında görünmektedir.
- Clip art seçeneğinden Google'ın sağladığı ikonlardan biri seçilip diğer seçeneklerden istenilen düzenleme yapılabilir.
- Daha sonra Androidmanifest.xml dosyasından istenilen icon eklenir.

KAYNAKÇA

- https://thinhme.wordpress.com/2015/10/24/lesson-6layout-in-android/
- https://stackoverflow.com/questions/3482742/what-is-the-difference-between-gravity-and-layout-gravity-in-android
- https://stackoverflow.com/questions/4619899/difference-between-a-views-padding-and-margin
- https://kiliccambaz.medium.com/android-farkl%C4%B1-ekran-boyutlar%C4%B1-i%CC%87le-%C3%A7al%C4%B1%C5%9Fma-f326cb2829e5