Bu yazımda setOnApplyWindowInsetsListener metodundan biraz bahsetmek istedim.Bu metod Android platformunda bir View sınıfının bir parçası olarak kullanılan önemli bir yöntemdir. Bu yöntem, bir View'in pencere kenar boşluklarına (window insets) uygulanan değişiklikleri dinlemek ve buna yanıt olarak görünümün boyutunu veya düzenini ayarlamak için kullanılır. Özellikle, bu yöntem, Android cihazların ekranında çentik, kenar boşlukları veya sistem çubuğu gibi ekran ögeleri olduğunda çok kullanışlıdır.Çünkü günümüzün tasarım kalıpları gittikçe buna evrilmektedir.

Bu metodun işlevini daha iyi anlamak için birkaç önemli konuya daha değinmemiz gerekiyor:

Window Insets Nedir?

Window insets, bir Android cihazının ekranında sistem tarafından uygulanan belirli bir alanı ifade eder. Örneğin, ekranın üst kısmında bir durum çubuğu veya alt kısmında bir gezinme çubuğu olduğunda, bu çubuklar ekranın içeriğini kaplayacak ve içeriğin bu alanlara yerleşmesini gerektirecektir.

setOnApplyWindowInsetsListener Metodu Ne Yapar?

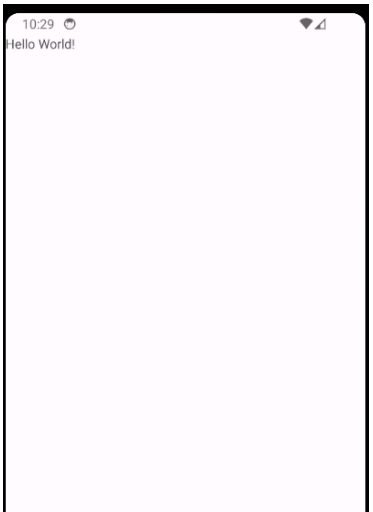
Bu metod, bir View'e bir pencere kenar boşluğu (window insets) uygulandığında çağrılır. Bu, özellikle, bir cihazın ekranında sistem çubukları, çentikler veya diğer kenar boşlukları gibi unsurlar olduğunda ortaya çıkar. Bu metod, View'in yeni boyutunu veya düzenini ayarlamak için kullanılabilir.Bu arada önemli bir hususu atlamak istemem.View derken illaki bir ekrandan bahsetmek istemem.Bir butonun da boşluklarını dinleyebiliriz tabiki.

Kullanım Alanları:

Pencere Kenar Boşluklarını Dinleme: Bu metod, pencere kenar boşluklarının değişikliklerini dinlemek için kullanılır. Örneğin, bir görünümün alt kısmında bir gezinme çubuğu olduğunda ve bu çubuğun görünümün boyutunu etkilemesi gerekiyorsa, bu metodu kullanarak bu değişikliklere yanıt verilebilir. ViewCompat.setOnApplyWindowInsetsListener metodu, bir View'e pencere kenar boşlukları (system insets) uygulandığında bir dinleyici tanımlar. Bu kod parçası, findViewById(R.id.main) ile belirtilen View üzerinde bu dinleyiciyi ayarlar. Yani, main id'sine sahip bir View varsa, onun pencere kenar boşluklarını dinler.

Ekran İçeriğini Ayarlama: Window insets, ekranın içeriğinin belirli bölgelere sığmasını gerektiriyorsa, bu metod kullanılarak View'in boyutu veya düzeni ayarlanabilir. Kod bloğu içindeki lambda ifadesi, pencere kenar boşluklarını alır ve systemBars adlı bir değişkende saklar. Ardından, bu kenar boşluklarını View'in padding'ine (kenar boşluğuna) uygular. Bu durumda, ekranın her bir kenarı için ayrı ayrı kenar boşlukları ayarlanır.

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
    val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
    insets ^setOnApplyWindowInsetsListener
}
```



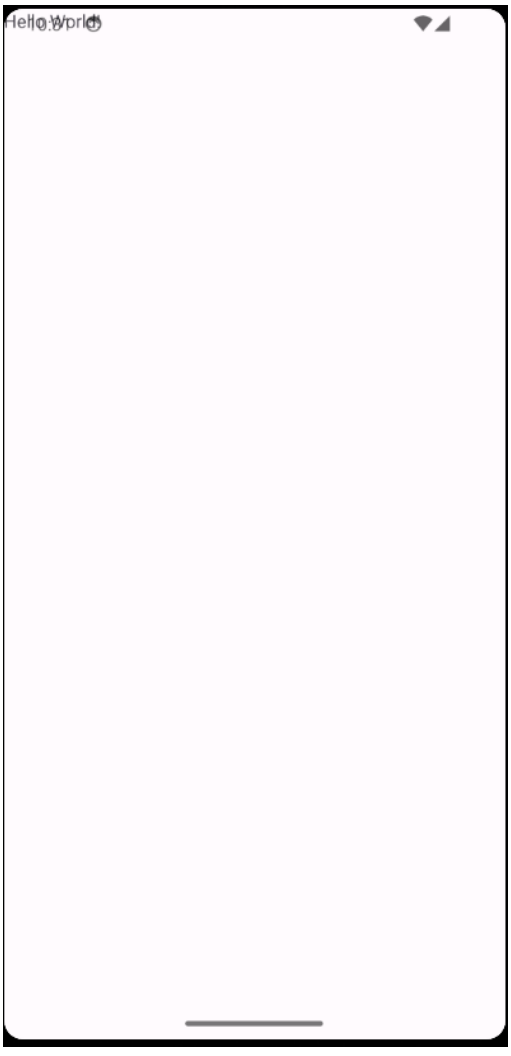
Default gelen hali bizi durum çubuğundan kurtarma hareketini yapar.

Bu sayede eklediğimiz elemanlar belirli bir yerden başlayarak,kullanılabilir olur.

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
    val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding( left: 200,  top: 200, systemBars.right, systemBars.bottom)
    insets ^setOnApplyWindowInsetsListener
}
```

Burada bu boşlukları kendimiz ayarlayıp istediğimiz kadar boşluk verebiliyoruz.

```
/* ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
    val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
    v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
    insets
}*/
```



Bunları yorum satırına aldığımızda da oluşabilecek sıkıntı yandadır.