

Mobil Uygulama Programlama

Dr.Öğ.Üyesi Ömer ÇETİN

Konu-6

Bu Dersten Önce

- Mobil uygulama programlama nedir? Neden ihtiyaç duyarız?
- Web programlama ile farkları nedir? Ne gibi avantajlar sağlar?
- Android uygulaması nedir?
- Android uygulama geliştirme ortamları nelerdir? Nasıl sağlanır?
- Android Studio IDE'sinin tanıtılması ve ilk proje geliştirilmesi ve derlenmesi
- Android Studio ile geliştirilen projenin bir mobil cihaz ve sanal makine üzerinde çalıştırılması
- Uygulama konfigürasyonu ve manifest dosyası
- Aktiviteler, aktivite durumları, yeni aktivite oluşturma
- Intent, Service, Broadcast, Content Providers, Resources (kaynaklar)
- Gradle
- Ara yüz geliştirme - Android layout türleri (Linear Layout - Relative Layout - Frame Layout - Grid Layout) / Dinamik Android Layout Oluşturma

Dr.Öğ.Üyesi Ömer ÇETİN

Bu Derste

- Android UI Bileşenleri
 - TextView ve AutoCompleteTextView
 - Button
 - EditText

Dr.Öğ.Üyesi Ömer ÇETİN

TextView

TextView yazıyı kullanıcıya göstermek amacıyla kullanılır. Android uygulamamızda bir çok yerde TextView kullanabiliriz, şimdi özelliklerini incelemeye başlayalım.

- **textColor**: Yazının rengini değiştirmek için kullanılır.
- **textSize**: Yazı font boyutunu belirlemek için kullanılır. **sp** (scale independent pixels) olarak belirlenir. Yazı boyutları tanımlamaları için sp kullanımı tavsiye edilmektedir. Ayrıca, yazı boyutunda **12sp** den daha küçük boyutlandırılmalardan kaçınılması android tarafından tavsiye edilir.
- **text**: Yazı da görükmesini istediğiniz metni tutar.
- **background**: Yazının arka plan rengini belirlemek için kullanılır.
- **alpha**: Yazının şeffaflığını ayarlamak için kullanılır.

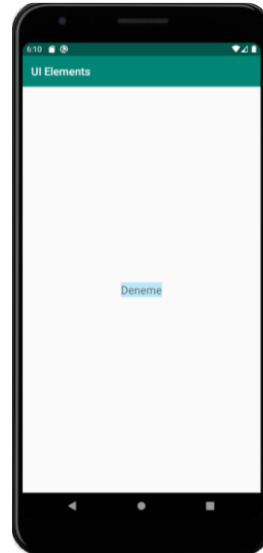
Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:alpha="0.6"
        android:background="#8edbf0"
        android:text="Deneme"
        android:textColor="#000"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

TextView

- **textAllCaps**: true olması durumunda yazıların tümü büyük harf ile oluşturulur.
- **hint**: TextView in text değeri boş olduğunda hint de yazan yazı gözüktür.
- **typeface**: (normal, sans, serif, monospace) tiplerinden birini yazının türü için kullanabilirsiniz.

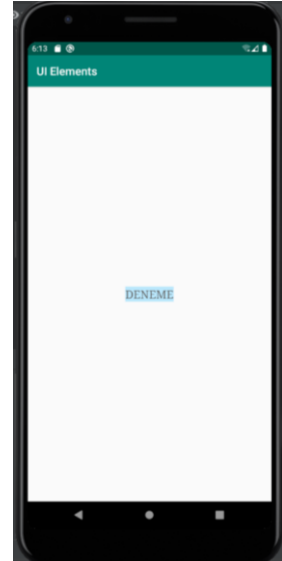
Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:alpha="0.6"
        android:background="#8edbf0"
        android:textAllCaps="true"
        android:typeface="serif"
        android:text="Deneme"
        android:textColor="#000"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

TextView

- **textStyle**: (normal , italic ,bold) tiplerinden birini yazının stili için kullanabilirsiniz. italic ile yazıyı eğimli bir şekilde gösterebilirsiniz, bold ile de yazıyı daha kalın gösterebilirsiniz.
- **singleLine**: true dediğimizde bir yataya göre kendini şekillendirir eğer uzun bir yazı gelirse bir alt satıra kalan yazılar geçmez tek bir satırda kalır ve sonuna üç nokta otomatik olarak konumlanır.

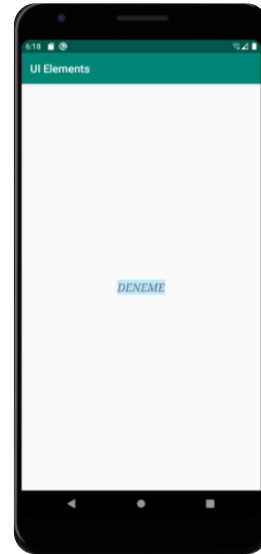
Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:alpha="0.6"
        android:background="#8edbf8"
        android:textAllCaps="true"
        android:typeface="serif"
        android:textStyle="italic"
        android:text="Deneme"
        android:textColor="#000"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:alpha="0.6"
        android:background="#8edbf8"
        android:textAllCaps="true"
        android:typeface="serif"
        android:textStyle="italic"
        android:text="Lorem ipsum dolor sit amet, debet tacimates quip, per id solet lab
        ores, Annagham nulla ma iorum is, e ium viris vocibus. Eos at esse graece
        mollis, per ceteros facilis ex. Sed stet saperet phaedrum te, an pri
        mutut nulla. Quo ne suas essent, id suscipit accusamus pro"
        android:singleLine="true"
        android:textColor="#000"
        android:textSize="20sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

TextView

- TextView de yer alan yazının otomatik olarak kaymasını istiyorsanız aşağıdaki gibi bir yapı kurmalısınız; **marqueeRepeatLimit** ile kayma hızını oluşturuyoruz, **singleLine** true burada da kullandık çünkü tek satırda olması gerekiyor yatay da otomatik olarak kayma yapılabilmesi için. Daha sonra yatay şekilde scroll modunu açabilmemiz için de **scrollHorizontally** true olarak set ettik.

Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:background="#8ed8fe"
        android:ellipsize="marquee"
        android:focusable="true"
        android:focusableInTouchMode="true"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true"
        android:singleLine="true"

        android:text="Lorem ipsum dolor sit amet, debet accipiamus quibus, per id soleat labore
        s. Amagnanulla mauris, eleum viris vocibus. Eos at esse graece mollis,
        per ceteros facilisi ex. Sed stet saperet phaedrum te, an pri mutat nulla.
        Quo ne suas essent, id suscipit accusamus pro"
        android:textColor="#0000"
        android:textSize="20sp"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="341dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



TextView

- **drawableTop**: TextView in yukarisina resim konumlandirmamizi saglar.
- **drawableBottom**: TextView in altina resim konumlandirmamizi saglar.
- **drawableRight**: TextView in sagina resim konumlandirmamizi saglar.
- **drawableLeft**: TextView in soluna resim konumlandirmamizi saglar.

Dr.Öğ.Üyesi Ömer ÇETİN

TextView

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:background="@#8edcfe"
        android:ellipsize="marquee"
        android:focusable="true"
        android:focusableInTouchMode="true"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true"
        android:singleLine="true"
        android:drawableLeft="@mipmap/ic_launcher"
        android:text="Deneme"
        android:textColor="@#000"
        android:textSize="20sp"
        tools:layout_editor_absoluteX="0dp"
        tools:layout_editor_absoluteY="341dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

AutoCompleteTextView Kullanımı

- Adında da anlaşılacağı gibi otomatik olarak tanımlayan textview türü aslında EditText den türetilerek oluşturulmuştur.
- Öncelikle **AutoCompleteTextView** bileşenini XML layout'da tanımlarız.
- Daha sonra ilgili activity de kullanıcı klavyeden giriş yaptığı zaman tamamlanmasını istediğimiz değerleri **Adapter** oluşturarak ayarlanır.
- Ekran çıktısına baktığınızda "Türkiye" yazmaya başladığında ortaya çıkan pop-up'da aranan kelime çıkar ve eğer aranan kelime doğru ise önerilen bu metne tıklayıp otomatik olarak ilgili alanın bu değeri alması sağlanır.

Dr.Öğ.Üyesi Ömer ÇETİN

AutoCompleteTextView Kullanımı

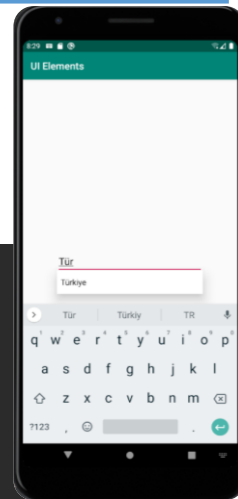
```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <AutoCompleteTextView
        android:id="@+id/txt"
        android:layout_centerInParent="true"
        android:textSize="20sp"
        android:textColor="#000000"
        android:popupBackground="#FFFFFF"
        android:hint="Aramak istediğiniz ülkeyi yazınız"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>

package com.example.uelements;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;

public class MainActivity extends AppCompatActivity {
    private static final String[] COUNTRIES = new String[] {
        "Türkiye", "Almanya", "Fransa", "Ukrayna", "İtalya"
    };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(getApplicationContext(),
            android.R.layout.simple_spinner_dropdown_item, COUNTRIES);
        AutoCompleteTextView textView = (AutoCompleteTextView)
            findViewById(R.id.txt);
        textView.setAdapter(adapter);
    }
}
```



Bu Derste

- Android UI Bileşenleri
 - TextView ve AutoCompleteTextView
 - **Button**
 - EditText
 - ImageView
 - Toggle Button, Checkbox, Switch ve RadioButton

Dr.Öğ.Üyesi Ömer ÇETİN

Button

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Giriş Yap" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

Button

Tıklama Olayları

- Yandaki kodlama gördüğünüz gibi butona **click eventi** vererek tıklama olayındaki işlemleri gerçekleştirebiliriz.

```
package com.example.uielements;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {
    private Button btnn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnn = (Button) findViewById(R.id.button_id);
        btnn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // buton tıklama anında çalışacak kısım
            }
        });
    }
}
```

Dr.Öğ.Üyesi Ömer ÇETİN

Button

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginEnd="143dp"
        android:layout_marginBottom="483dp"
        android:text="Giriş Yap" />

    <Button
        android:id="@+id/button_id2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_marginStart="179dp"
        android:layout_marginTop="98dp"
        android:layout_marginEnd="136dp"
        android:layout_marginBottom="585dp"
        android:text="Giriş Yap2" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

Button - Tıklandığında

```
package com.example.uelements;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private Button btnn;
    private Button btnn2;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnn = (Button) findViewById(R.id.button_id);
        btnn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // buton tıklama anında çalışacak kısım
                Toast.makeText(getApplicationContext(), "Deneme", Toast.LENGTH_LONG).show();
            }
        });

        btnn2 = (Button) findViewById(R.id.button_id2);
        btnn2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // buton tıklama anında çalışacak kısım
                Toast.makeText(getApplicationContext(), "Deneme2", Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

Dr.Öğ.Üyesi Ömer ÇETİN

OnClickListener methodunu implement ettiğimizde onClick eventini tanımlamamızı isteyecektir. Bu şekilde tanımlama yaptıktan sonra tıklama olayını tetikleyebiliriz.



Button – Uzun Tıklandığında

- Normal tıklama olayının kullanım şekillerini gösterdik şimdi de **OnLongClickListener** ile uzun tıklama olayının nasıl gerçekleştiğine bakalım.

```
package com.example.uelements;

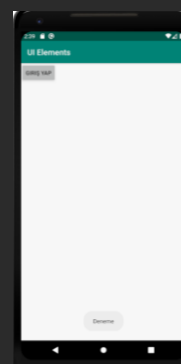
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private Button btnn;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        btnn = (Button) findViewById(R.id.button_id);
        btnn.setOnLongClickListener(new View.OnLongClickListener() {
            @Override
            public boolean onLongClick(View v) {
                Toast.makeText(getApplicationContext(), "Deneme", Toast.LENGTH_LONG).show();
                return false;
            }
        });
    }
}
```

Dr.Öğ.Üyesi Ömer ÇETİN



Button – Resim ile Birlikte

- Buton içeriğimizin sol tarafına bir imaj konumlandırmak istersek yapmamız gerekenler;

- `android:drawableLeft`
- `android:drawableRight`

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:drawableLeft="@mipmap/ic_launcher"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Giriş Yap" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Butonumuzun içindeki text in rengini (textColor) ve buton arka plan rengini (background) ayarlayalım;

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:background="#0000FF"
        android:textColor="#FFF"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Giriş Yap" />

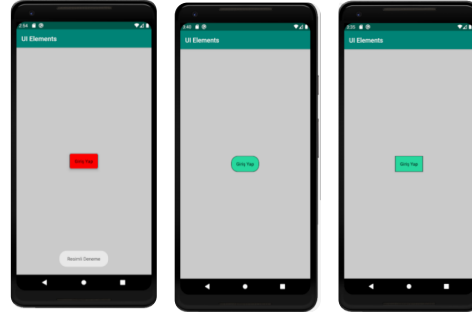
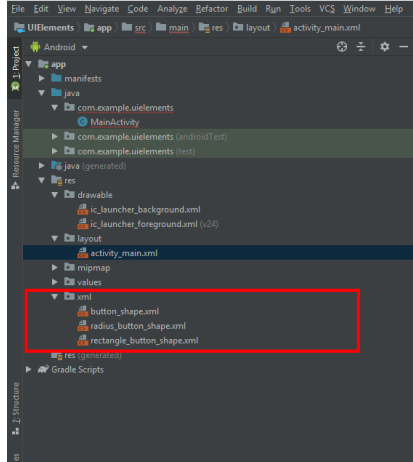
</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Özel Buton Formları Oluşturma



Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Butona tıklandığı anda **tıklama efekti** nasıl verebiliriz bakalım: öncelikle **res->XML** klasörüne aşağıdaki XML dosyası oluşturalım. Böylece butonun ilk yaratıldığındaki rengini ve biçimini belirledik ayrıca butonun tıklama anındaki rengini ve biçimini de belirledik.

button_shape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android" >
  <item android:state_pressed="true" >
    <shape android:shape="rectangle" >
      <corners android:radius="5dp" />
      <solid android:color="#FF0000" />
      <stroke android:width="1dp" android:color="#0dbcbf" />
    </shape>
  </item>
  <item>
    <shape android:shape="rectangle" >
      <corners android:radius="5dp" />
      <solid android:color="#0dbcbf" />
    </shape>
  </item>
</selector>
```

Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Yukarıda oluşturduğumuz button_shape dosyasını butonumuza arka plan olarak set ediyoruz.

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ccc"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:textAllCaps="false"
        android:background="@drawable/button_shape"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Giriş Yap" />

</RelativeLayout>
```

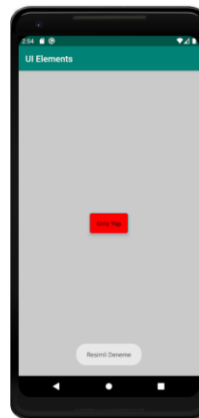
Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Bu şekilde elde ettiğimiz ekran görüntüsü aşağıdadır (butona **tıklanmadan önceki** görünüm)
- Bu şekilde elde ettiğimiz ekran görüntüsü aşağıdadır (butona **tıklandığı esnada** oluşan görünüm)



Butona
tıklandığında
button_shape
sayesinde
oluşturduğumuz
tıklama efekti ile
buton kırmızı oluyor
ve gerçekten
tıklanma hissini elde
etmiş oluyoruz.



Dr.Öğ.Üyesi Ömer ÇETİN

Buton – Özelleştirmeler

- Dikdörtgen bir buton elde etmek istersek drawable klasörüne aşağıdaki xml

rectangle_button_shape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <solid android:color="#27da9f" />

    <stroke
        android:width="1dp"
        android:color="#000" />

</shape>
```

activity_main.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ccc"
    android:orientation="vertical" >

    <Button
        android:id="@+id/button_id"
        android:textAllCaps="false"
        android:background="@xml/rectangle_button_shape"
        android:layout_centerInParent="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Giriş Yap" />

</RelativeLayout>
```

Dr.Öğ.Üyesi Ömer ÇETİN



Buton – Özelleştirmeler

- Butonun kenar yarı çapına göre bir görüntü elde etmek istersek XML klasörüne aşağıdaki xml dosyasını oluşturduk.

radius_button_shape.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

    <corners
        android:bottomLeftRadius="20dp"
        android:bottomRightRadius="20dp"
        android:radius="1dp"
        android:topLeftRadius="20dp"
        android:topRightRadius="20dp" />

    <solid android:color="#27da9f" />

    <stroke
        android:width="1dp"
        android:color="#000" />

</shape>
```

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle" >

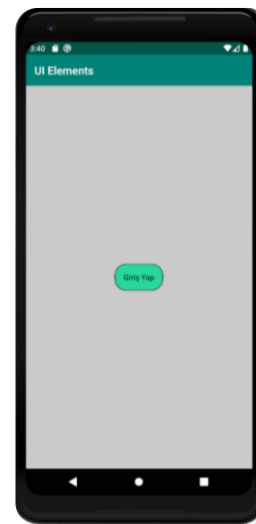
    <corners
        android:bottomLeftRadius="30dp"
        android:bottomRightRadius="20dp"
        android:radius="1dp"
        android:topLeftRadius="20dp"
        android:topRightRadius="20dp" />

    <solid android:color="#27da9f" />

    <stroke
        android:width="1dp"
        android:color="#000" />

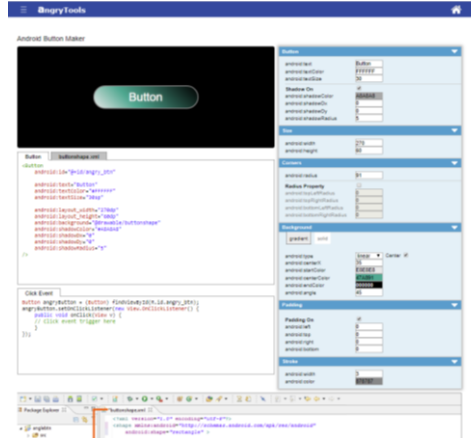
</shape>
```

Dr.Öğ.Üyesi Ömer ÇETİN



Button – Estetik Buton Hazırlama

- Örnek Online Uygulama: angryTools
- <http://angrytools.com/android/button/>



Dr.Öğ.Üyesi Ömer ÇETİN

Bu Derste

- Android UI Bileşenleri
 - TextView ve AutoCompleteTextView
 - Button
 - **EditText**
 - ImageView
 - Toggle Button, Checkbox, Switch ve RadioButton

Dr.Öğ.Üyesi Ömer ÇETİN

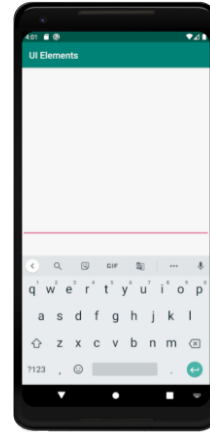
EditText

EditText layout xml kısmında tanımlamasını yapalım;

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/txtInput"
        android:layout_centerInParent="true"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

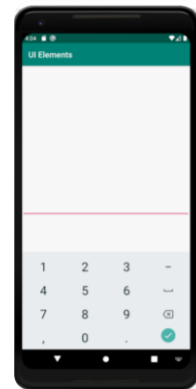
EditText

- EditText için girilebilecek tipleri **inputType** ile belirliyoruz. Sonuçta kullanıcının o alana sadece sayısal değer girmesini isteyebiliriz ya da sadece metin girmesini isteyebiliriz:
- **inputType="text"** kullanımı ile metin değerlerinin girilmesini sağlarız.
- **inputType="number"** kullanımı ile sadece sayısal değerlerin girilmesini sağlarız.
- **inputType="textPassword"** kullanımı ile metin değerlerinin yıldızlı bir şekilde gözükmeden girilmesini sağlarız.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/txtInput"
        android:inputType="number"
        android:layout_centerInParent="true"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        />

</RelativeLayout>
```

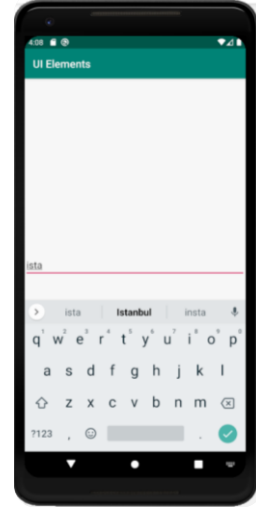


Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- **inputType="textCapSentences|textAutoCorrect"**

kullanımı ile metin cümlesi girilmesi ve girilen değere göre bize otomatik olarak metin önerileri sunmayı sağlar. (Cihaz diline göre önerileri sunar ve otomatik doğrulama için yine dil seçenekleri dikkat alınmaktadır.)



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

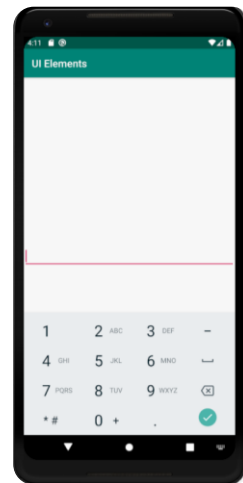
- **inputType="phone"** kullanımı ile telefon numarasına özgün değerlerin girilmesini sağlarız.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/txtInput"
        android:inputType="phone"
        android:layout_centerInParent="true"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        />

</RelativeLayout>
```

Yukarıda örnekleri verildiği gibi bir çok veri tipi mevcuttur. EditText için kullanılış açısından bir kaç tanesini gösterilmiştir, diğer tipleri kendinizde inceleyebilirsiniz.



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- res->values->styles.xml dosyasına girip EditText için kullanacağımız bir style oluşturuyoruz aşağıdaki kodlama gördüğünüz gibi **colorAccent** adındaki değere renk kodunu atadık.

CustomEditTextStyle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="CustomEditTextStyle" parent="Widget.AppCompat.EditText">
        <item name="colorAccent">#3F51B5</item>
    </style>
</resources>
```

- Oluşturduğumuz **style** dosyasını EditText bileşenine tema olarak veriyoruz. Yukarıda oluşturduğumuz style dosyasının adı ne ise onu EditText e vermeliyiz. Biz adını **CustomEditTextStyle** olarak oluşturduk sizde bu kısmı istediğiniz gibi adlandırabilirsiniz.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <EditText
        android:id="@+id/txtInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        />

</RelativeLayout>
```

Dr.Öğ.Üyesi Ömer ÇETİN



EditText

- XML layout da yer alan EditText e **findViewById** yardımı ile erişiyoruz böylece eriştiğimiz Android bileşenini direk olarak kullanabilmeyi sağlıyoruz.

- **setText()**: EditText e değer girilmesini sağlar.

MainActivity.java

```
package com.example.uielements;

import ...

public class MainActivity extends AppCompatActivity {

    private EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = (EditText) findViewById(R.id.txtInput);
        editText.setText("Ömer ÇETİN");
    }
}
```

Dr.Öğ.Üyesi Ömer ÇETİN



EditText

```
package com.example.uielements;

import ...

public class MainActivity extends AppCompatActivity {

    private EditText editText;
    private EditText editText2;
    private EditText editText3;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = (EditText) findViewById(R.id.txtInput);
        editText.setText("EditText1");

        String str = editText.getText().toString();

        editText2 = (EditText) findViewById(R.id.txtInput2);
        editText2.setText("EditText2");

        editText3 = (EditText) findViewById(R.id.txtInput3);
        editText3.setText(str);
        editText3.selectAll();
    }
}
```

Dr.Öğ.Üyesi Ömer ÇETİN

- **selectAll():** EditText içindeki değerin seçili bir şekilde gelmesini sağlar.
- **getText():** EditText içinde yer alan Editable metin değerini döndürmeyi sağlar.
- **setSelection():** EditText içinde yer alan değerin belirtilen index aralığına göre seçili gelmesini sağlar.

EditText

- **maxLines:** EditText için maksimum satır sayısını belirtir. maxLines değerini 1 olarak belirttik. Genelde **android:singleLine="true"** olarak da görebilirsiniz fakat deprecated (artık kullanımı önerilmeyen) olduğu için onun yerine **maxLines** kullandık. Tek satır yer kaplasın dediğimiz için aşağıdaki gibi uzun bir metin yer aldığında parmağımızla kaydırma işlemi yaparak sığmayan diğer metin kısımlarını görebiliriz.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >

<EditText
android:id="@+id/textInput"
android:layout_centerInParent="true"
android:theme="@style/CustomEditTextStyle"
android:text="Lorem ipsum dolor sit amet, in nec alii dolorem
offendit, qui ludus zril epicurei et. Modus minim impedit eam ex, porto
temporibus concludaturque ne sit, ceteros molestiae sit ne."
android:maxLines="1"
android:inputType="text"
android:layout_width="match_parent"
android:layout_height="wrap_content" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

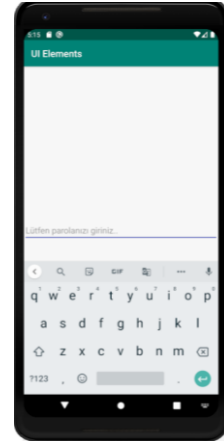
EditText

- **hint:** İpucu girilmesini sağlar. Kullanıcıya yol göstermek için de kullanılır. Örneğin; belirtilen EditText de ne kullanıcının o alana ne yazması gerektiğini hint yardımı ile gösterebiliriz. EditText içindeki değer kullanıcı tarafından yazılmaya başlandığı an hint değeri kaybolur , sıfırlandığı an ise geri gözükür.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/txtInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:hint="Lütfen parolanızı giriniz.."
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- **textColor:** EditText içindeki değerini belirlemek için kullanılır.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/txtInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:textColor="#c70606"
        android:text="Android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- **drawableRight:** EditText in sağ tarafına imaj koymamızı sağlar.

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/txtInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:drawableRight="@drawable/search_icon"
        android:text="Android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- EditText focus olduğunda bazı işlemlerin gerçekleşmesi için **onFocusChangeListener** kullanabiliriz.

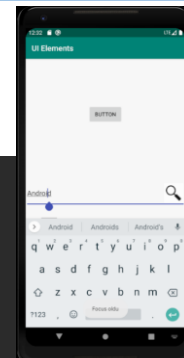
MainActivity.java

```
package com.example.uielements;
import ...
public class MainActivity extends AppCompatActivity {

    private EditText editText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        editText = (EditText) findViewById(R.id.txtInput);
        editText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
            @Override
            public void onFocusChange(View view, boolean hasFocus) {
                if (hasFocus) {
                    Toast.makeText(getApplicationContext(), "Focus oldu", Toast.LENGTH_LONG);
                } else {
                    Toast.makeText(getApplicationContext(), "Focus kaybettii", Toast.LENGTH_LONG);
                }
            }
        });
    }
}
```



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- EditText bileşenini dikdörtgen şeklinde yapmak istersek aşağıdaki gibi drawable klasörü içine bir XML dosyası oluşturmalıyız.

border_edittext.xml

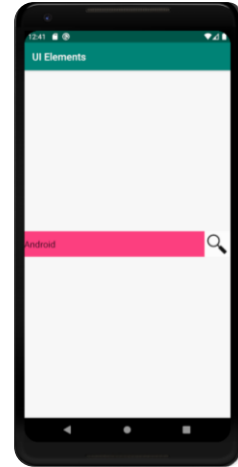
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FF4081"/>
</shape>
```

activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/textInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:background="@drawable/border_edittext"
        android:drawableRight="@drawable/search_icon"
        android:text="Android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- Eğer EditText bileşenini köşelerden hafif oval şeklinde yapmak istersek aşağıdaki gibi drawable klasörü içine bir xml dosyası oluşturmalıyız.

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FF4081"/>
    <corners
        android:bottomRightRadius="15dp"
        android:bottomLeftRadius="15dp"
        android:topLeftRadius="15dp"
        android:topRightRadius="15dp"/>
</shape>
```

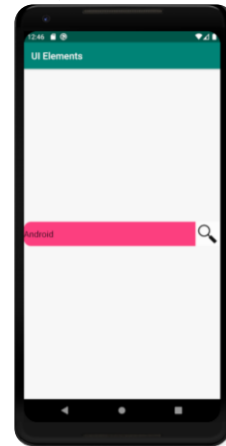
border_edittext.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/textInput"
        android:layout_centerInParent="true"
        android:theme="@style/CustomEditTextStyle"
        android:background="@drawable/border_edittext"
        android:drawableRight="@drawable/search_icon"
        android:text="Android"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

activity_main.xml



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- Material Design ile yaygınlaşan EditText in TextInputLayout ile birlikte kullanımı da aşağıdaki gibidir. Normal EditText bileşeni üstte altında da TextInputLayout ile kullanılan EditText yer almaktadır. Bu şekilde kullanımda o alana focus olduğunda hint değeri animasyonlu olarak üst kısma çıkmaktadır zaten iki EditText değerini bir arada kullanırsanız farkı anlayacaksınız.
- not: Aşağıdaki şekilde TextInputLayout ile EditText i birlikte kullanmak istiyorsanız support design kütüphanesinin de ekli olması gerekmektedir. Android projenizin build.gradle dosyası içinde dependencies kısmında bu kodun yer aldığından emin olunuz !

implementation 'com.android.support:design:23.0.1'

Dr.Öğ.Üyesi Ömer ÇETİN

EditText

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:orientation="vertical"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/edittext"
            android:hint="Lütfen adınızı giriniz.."
            android:layout_width="match_parent"
            android:layout_height="wrap_content" />

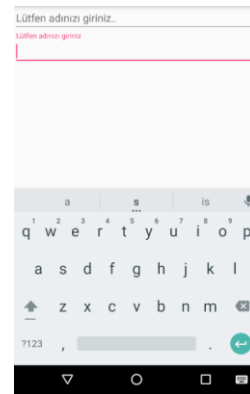
        <android.support.design.widget.TextInputLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content">

            <EditText
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:hint="Lütfen adınızı giriniz" />

        </android.support.design.widget.TextInputLayout>

    </LinearLayout>
</RelativeLayout>
```

activity_main.xml



Dr.Öğ.Üyesi Ömer ÇETİN

EditText

- EditText üzerinde herhangi bir işlemden dolayı kullanıcıya hata mesajı göstermek istersek aşağıdaki gibi setError metodunu kullanabiliriz.

MainActivity.java

```
package com.example.uitablements;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
import android.view.View;  
import android.widget.EditText;  
  
public class MainActivity extends AppCompatActivity {  
  
    private EditText editText;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        editText = (EditText) findViewById(R.id.txtInput);  
        editText.setError("Kullanıcı adı hatalı");  
    }  
}
```



Dr.Öğ.Üyesi Ömer ÇETİN