

Stilovi i teme

Mobile App Development 11

Nikola Đukić

19. oktobar 2020.

Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 CardView
- 4 RecyclerView

Stilovi

- Stil je skup svojstava koja specificiraju izgled određenog pogleda

Stilovi

- Stil je skup svojstava koja specificiraju izgled određenog pogleda
- Neka od tih svojstava su visina (height), širina (width), punjene (padding), margina (margin), font (typeface), boja teksta (font color), veličina teksta (font size), boja pozadine (background color), itd.

main_activity.xml

```
1  <TextView
2      android:id="@+id/text_view"
3      android:text="@string/hello_world"
4      android:layout_width="fill_parent"
5      android:layout_height="wrap_content"
6      android:typeface="monospace"
7      android:textSize="15sp"
8      android:textStyle="italic"
9      android:textColor="#00FF00" />
```

main_activity.xml

```
1  <TextView
2      android:id="@+id/text_view"
3      android:text="@string/hello_world"
4      android:layout_width="fill_parent"
5      android:layout_height="wrap_content"
6      style="@style/example_style" />
```

style.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <style name="example_style">
4          <item name="android:typeface">monospace</item>
5          <item name="android:textSize">15sp</item>
6          <item name="android:textStyle">italic</item>
7          <item name="android:textColor">#00FF00</item>
8      </style>
9  </resources>
```

Stilovi

- Stilovi se definišu kao poseban resurs (odgovara im XML dokument u res/values direktorijumu, i obicno se zove styles.xml)

Stilovi

- Stilovi se definišu kao poseban resurs (odgovara im XML dokument u res/values direktorijumu, i obicno se zove styles.xml)
- Dokument sa stilovima sadrži ključ-vrednost parove

Stilovi

- Stilovi se definišu kao poseban resurs (odgovara im XML dokument u res/values direktorijumu, i obicno se zove styles.xml)
- Dokument sa stilovima sadrži ključ-vrednost parove
- Stilove je moguće nasleđivati

style.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <resources>
3      <style name="example_style" parent="@android:style/Theme.Material">
4          <item name="android:typeface">monospace</item>
5          <item name="android:textSize">15sp</item>
6          <item name="android:textStyle">italic</item>
7          <item name="android:textColor">#00FF00</item>
8      </style>
9  </resources>
```

Teme

- Tema je skup atributa koji se primenjuje na citavu aplikaciju ili aktivnost umesto na pojedinačan pogled

Teme

- Tema je skup atributa koji se primenjuje na citavu aplikaciju ili aktivnost umesto na pojedinačan pogled
- Tada se za svaki pogled aplikacije ili aktivnosti primenjuje svako svojstvo teme koje pogled podržava

Teme

- Tema je skup atributa koji se primenjuje na citavu aplikaciju ili aktivnost umesto na pojedinačan pogled
- Tada se za svaki pogled aplikacije ili aktivnosti primenjuje svako svojstvo teme koje pogled podržava
- Tema predstavlja skup imenovanih resursa koji se kasnije mogu referncirati iz stilova, rasporeda... (npr. tema pruža boju `colorPrimary`, koja se može koristiti u stilovima)

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest ... >
3      <application android:theme="@android:style/Theme.Material" ... >
4          <!-- ... -->
5      </application>
6  </manifest>
```

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest ... >
3      <application ... >
4          <activity
5              android:name=".ExampleActivity"
6              android:theme="@android:style/Theme.Material" ... >
7              <intent-filter>
8                  <action android:name="android.intent.action.MAIN" />
9                  <category android:name="android.intent.category.LAUNCHER" />
10             </intent-filter>
11         </activity>
12     </application>
13 </manifest>
```


Pregled sadržaja

- 1 Stilovi i teme
- 2 Material Design
- 3 CardView
- 4 RecyclerView

Material Design

- Material Design je skup principa za vizuelni dizajn, dizajn pokreta i dizajn interakcija

Material Design

- Material Design je skup principa za vizuelni dizajn, dizajn pokreta i dizajn interakcija
- Aplikacije dizajnirane po ovim principima pružaju korisnicima konzistentno iskustvo na različitim platformama (mobilnim, web i desktop) i u različitim aplikacijama

Material Design

- Material Design je skup principa za vizuelni dizajn, dizajn pokreta i dizajn interakcija
- Aplikacije dizajnirane po ovim principima pružaju korisnicima konzistentno iskustvo na različitim platformama (mobilnim, web i desktop) i u različitim aplikacijama
- Material Design koristi metafore da bi korisničko iskustvo bilo intuitivno

Material Design

Principi Material Design preporuka mogu se grupisati u tri kategorije:

- opipljive površine (tengable sufraces)

Material Design

Principi Material Design preporuka mogu se grupisati u tri kategorije:

- opipljive površine (tengable sufraces)
- smeo grafički dizajn (bold graphic design)

Material Design

Principi Material Design preporuka mogu se grupisati u tri kategorije:

- opipljive površine (tengable sufaces)
- smeo grafički dizajn (bold graphic design)
- smisleni pokreti (meaningful motion)

Opipljive površine

Senke simuliraju visinu listova papira koja određuje njihov međusobni odnos:

- seam (dva lista papira koji dele zajedničku ivicu se kreću zajedno)

Opipljive površine

Senke simuliraju visinu listova papira koja određuje njihov međusobni odnos:

- seam (dva lista papira koji dele zajedničku ivicu se kreću zajedno)
- step (dva lista papira koji se preklapaju se kreću nezavisno)

Opipljive površine

Senke simuliraju visinu listova papira koja određuje njihov međusobni odnos:

- seam (dva lista papira koji dele zajedničku ivicu se kreću zajedno)
- step (dva lista papira koji se preklapaju se kreću nezavisno)
- floating action button (dugme odvojeno od toolbar-a)

Smeo grafički dizajn

Na listovima se prikazuje:

- tekst (Roboto i Noto)

Smeo grafički dizajn

Na listovima se prikazuje:

- tekst (Roboto i Noto)
- fotografije, ilustracije i ikonografija (predefinisane ikone za uobičajene akcije)

Smeo grafički dizajn

Na listovima se prikazuje:

- tekst (Roboto i Noto)
- fotografije, ilustracije i ikonografija (predefinisane ikone za uobičajene akcije)
- boje (primarna, sekundarna i akcentovana)

Smisleni pokreti

- autentični pokreti (pokreti treba da budu usklađeni sa masom, zapreminom i fleksibilnošću objekta)

Smisleni pokreti

- autentični pokreti (pokreti treba da budu usklađeni sa masom, zapreminom i fleksibilnošću objekta)
- interakcija sa kraktim odzivom (aplikacije reaguju na akcije korisnika i obezbeđuju vizuelnu potvrdu)

Smisleni pokreti

- autentični pokreti (pokreti treba da budu usklađeni sa masom, zapreminom i fleksibilnošću objekta)
- interakcija sa kraktim odzivom (aplikacije reaguju na akcije korisnika i obezbeđuju vizuelnu potvrdu)
- smisleni prelazi (prelazi treba da usmere pažnju korisnika i da budu glatki)

Material Design za Android

Android podržava Material Design tako što pruža:

- nove teme

Material Design za Android

Android podržava Material Design tako što pruža:

- nove teme
- nove poglede (npr. RecyclerView, CardView, itd.)

Material Design za Android

Android podržava Material Design tako što pruža:

- nove teme
- nove poglede (npr. RecyclerView, CardView, itd.)
- novi API za senke i animacije

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest ... >
3      <application android:theme="@android:style/Theme.Material" ... >
4          <!-- ... -->
5      </application>
6  </manifest>
```

Material Design teme

| Konstanta | Opis |
|------------------------------|------------------------------|
| Material | tamna verzija |
| Material.Light | svetla verzija |
| Material.Light.DarkActionBar | svetla verzija sa toolbar-om |

Tabela 1: Izabrane Material Design teme.

Reference

- Material Design, <https://material.io/design>

Reference

- Material Design, <https://material.io/design>
- Material Design for Android,
<http://developer.android.com/design/material>

CardView

- CardView predstavlja još jedan važan element uveden sa Material Design-om

CardView

- CardView predstavlja još jedan važan element uveden sa Material Design-om
- Upotrebom CardView-a možete prikazati informacije korisniku u vidu kartice

CardView

- CardView predstavlja još jedan važan element uveden sa Material Design-om
- Upotrebom CardView-a možete prikazati informacije korisniku u vidu kartice
- CardView predstavlja FrameLayout sa zaobljenim uglovima i senkom.

Upotreba CardView-a

- Dodati odgovarajuću CardView biblioteku u Gradle

Upotreba CardView-a

- Dodati odgovarajuću CardView biblioteku u Gradle
- Dodati CardView u odgovarajući raspored

build.gradle (app)

```
1 dependencies {  
2     implementation 'androidx.cardview:cardview:1.0.0'  
3 }
```

AndroidManifest.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.cardview.widget.CardView
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:tools="http://schemas.android.com/tools"
5      xmlns:app="http://schemas.android.com/apk/res-auto"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      app:cardElevation="8dp"
9      app:cardCornerRadius="16dp"
10     tools:context=".MainActivity">
11
12     <TextView
13         android:id="@+id/text_view"
14         android:layout_width="wrap_content"
15         android:layout_height="wrap_content"/>
16
17 </androidx.cardview.widget.CardView>
```

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView
- RecyclerView model predstavlja skup komponenti koje zajedno omogućavaju prikaz podakata

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView
- RecyclerView model predstavlja skup komponenti koje zajedno omogućavaju prikaz podakata
- Layout Manager komponenta obezbeđuje način na koji će se elementi prikazivati korisniku:

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView
- RecyclerView model predstavlja skup komponenti koje zajedno omogućavaju prikaz podakata
- Layout Manager komponenta obezbeđuje način na koji će se elementi prikazivati korisniku:
 - U vidu liste

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView
- RecyclerView model predstavlja skup komponenti koje zajedno omogućavaju prikaz podakata
- Layout Manager komponenta obezbeđuje način na koji će se elementi prikazivati korisniku:
 - U vidu liste
 - U vidu matrice

RecyclerView

- RecyclerView predstavlja fleksibilniji i napredniji ListView/GridView
- RecyclerView model predstavlja skup komponenti koje zajedno omogućavaju prikaz podakata
- Layout Manager komponenta obezbeđuje način na koji će se elementi prikazivati korisniku:
 - U vidu liste
 - U vidu matrice
 - U vidu smaknute matrice

ViewHolder

- ViewHolder objekat predstavlja pojedinačni element liste (RecyclerView-a)

ViewHolder

- ViewHolder objekat predstavlja pojedinačni element liste (RecyclerView-a)
- ViewHolder objekat je primerak klase koja nasleđuje klasu RecyclerView.ViewHolder

ViewHolder

- ViewHolder objekat predstavlja pojedinačni element liste (RecyclerView-a)
- ViewHolder objekat je primerak klase koja nasleđuje klasu RecyclerView.ViewHolder
- Svaki ViewHolder je zadužen za prikaz jednog elementa liste

ViewHolder

- ViewHolder objekat predstavlja pojedinačni element liste (RecyclerView-a)
- ViewHolder objekat je primerak klase koja nasleđuje klasu RecyclerView.ViewHolder
- Svaki ViewHolder je zadužen za prikaz jednog elementa liste
- RecyclerView pomoću ViewHolder pristupa omogućava bolje performanse u radu

Upotreba RecyclerView-a

- Dodati library za RecyclerView u gradle

Upotreba RecyclerView-a

- Dodati library za RecyclerView u gradle
- Dodati RecyclerView u raspored

Upotreba RecyclerView-a

- Dodati library za RecyclerView u gradle
- Dodati RecyclerView u raspored
- Kreirati RecyclerView adapter

Upotreba RecyclerView-a

- Dodati library za RecyclerView u gradle
- Dodati RecyclerView u raspored
- Kreirati RecyclerView adapter
- Povezati RecyclerView sa LayoutManager-om

Upotreba RecyclerView-a

- Dodati library za RecyclerView u gradle
- Dodati RecyclerView u raspored
- Kreirati RecyclerView adapter
- Povezati RecyclerView sa LayoutManager-om
- Povezati adapter i RecyclerView

build.gradle (app)

```
1 dependencies {  
2     implementation 'androidx.recyclerview:recyclerview:1.1.0'  
3 }
```

main_activity.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.recyclerview.widget.RecyclerView
3      android:id="@+id/recycler_view"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:scrollbars="vertical" />
```

RecyclerView.Adapter.java (1/2)

```

1  public class RecyclerView.Adapter<RecyclerView.ViewHolder>
2  {
3      private String[] mDataset;
4
5      // Provide a reference to the views for each data item
6      public static class ViewHolder extends RecyclerView.ViewHolder {
7          // each data item is just a string in this case
8          public TextView mTextView;
9
10         public ViewHolder(TextView v) {
11             super(v);
12             mTextView = v;
13         }
14     }
15     public RecyclerView.Adapter(String[] myDataset) {
16         mDataset = myDataset;
17     }
18     // Create new views (invoked by the layout manager)
19     @Override
20     public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
21         // create a new view
22         TextView v = (TextView) LayoutInflater.from(parent.getContext())
23             .inflate(R.layout.my_text_view, parent, false);
24         //...
25         ViewHolder vh = new ViewHolder(v);
26         return vh;
27     }

```


RecyclerView.Adapter.java (2/2)

```
1      // Replace the contents of a view (invoked by the layout manager)
2      @Override
3      public void onBindViewHolder(ViewHolder holder, int position) {
4          // - get element from your dataset at this position
5          // - replace the contents of the view with that element
6          holder.mTextView.setText(mDataset[position]);
7      }
8
9      // Return the size of your dataset (invoked by the layout manager)
10     @Override
11     public int getItemCount() {
12         return mDataset.length;
13     }
14 }
```

Povezivanje sa LayoutManager-om

```
1 public class MainActivity extends AppCompatActivity {  
2  
3     private RecyclerView recyclerView;  
4     private RecyclerView.Adapter adapter;  
5     private RecyclerView.LayoutManager layoutManager;  
6     @Override  
7     protected void onCreate(Bundle savedInstanceState) {  
8         super.onCreate(savedInstanceState);  
9         setContentView(R.layout.activity);  
10        recyclerView = (RecyclerView) findViewById(R.id.recycler_view);  
11        // use this setting to improve performance if you know that changes  
12        // in content do not change the layout size of the RecyclerView  
13        recyclerView.setHasFixedSize(true);  
14        // use a linear layout manager  
15        layoutManager = new LinearLayoutManager(this);  
16        recyclerView.setLayoutManager(layoutManager);  
17    }  
18 }
```

Povezivanje Adaptera i RecyclerView-a

```
1 public class MainActivity extends AppCompatActivity {  
2  
3     private RecyclerView recyclerView;  
4     private RecyclerView.Adapter adapter;  
5     private RecyclerView.LayoutManager layoutManager;  
6     @Override  
7     protected void onCreate(Bundle savedInstanceState) {  
8         super.onCreate(savedInstanceState);  
9         setContentView(R.layout.activity);  
10        recyclerView = (RecyclerView) findViewById(R.id.recycler_view);  
11        // use this setting to improve performance if you know that changes  
12        // in content do not change the layout size of the RecyclerView  
13        recyclerView.setHasFixedSize(true);  
14        // use a linear layout manager  
15        layoutManager = new LinearLayoutManager(this);  
16        recyclerView.setLayoutManager(layoutManager);  
17  
18        // specify an adapter and assign adapter  
19        // assume data has already assigned  
20        adapter = new RecyclerViewAdapter(data);  
21        recyclerView.setAdapter(adapter);  
22    }  
23 }
```



All images copyrighted by Android Open Source Project (CC BY)