

Build.gradle

BuildTypes

- debug, release + vlastní
- dokážeme v pomoci nich přizpůsobovat parametry při vytváření APK
- signingConfig
- proguard

```
release {  
    keyAlias "alias"  
    storeFile file("../key.keystore")  
    storePassword "abcdef"  
    keyPassword "123456"  
}
```

```
...
```

```
signingConfig signingConfigs.release
```

ProductFlavors

- varianty aplikace s různou “příchutí”
- různé res a/nebo Java kód pro každou variantu
- free/paid, test/live ...
- příklad!

Themes & Styles

- **style** - sada XML atributů ovlivňujících vzhled UI
 - velikosti, barvy, stíny, písma, ...
- **theme** - style aplikovaný na celou Activity
 - ovlivňuje všechny View, které nastavený XML atribut podporují
 - může být přiřazen i k View
- zajišťují konzistenci UI, urychlují vývoj

```
<resources>
```

```
<!-- Base application theme. -->
```

```
<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
```

```
<!-- Customize your theme here. -->
```

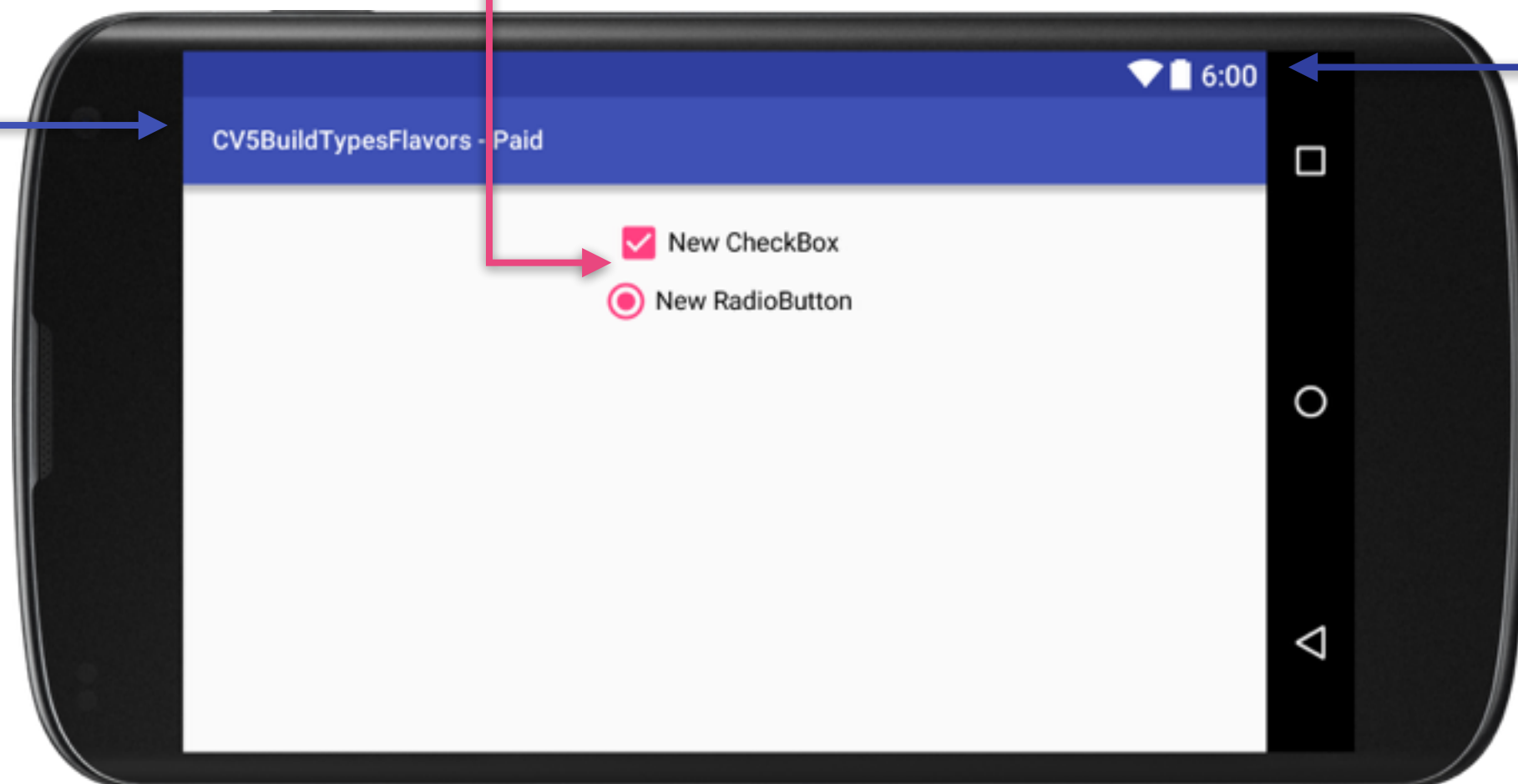
```
<item name="colorPrimary">@color/colorPrimary</item>
```

```
<item name="colorPrimaryDark">@color/colorPrimaryDark</item>
```

```
<item name="colorAccent">@color/colorAccent</item>
```

```
</style>
```

```
</resources>
```



- `Theme.AppCompat.Light.DarkActionBar` ?!?
- Theme lze přiřazovat jednotlivým aktivitám zvlášť pomocí xml tagu `android:theme`
- Spousta atributů umožňujících nastavit:
 - defaultní barvy, parametry okna, stíny aj.
 - kde je najdu?
- ukázka!

Vlastní styly



Hello world!

```
<TextView
```

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:background="@drawable/bg_rounded_red"  
    android:padding="20dp"  
    android:typeface="monospace"  
    android:textSize="20sp"  
    android:gravity="center"  
    android:textColor="@android:color/white"  
    android:layout_margin="10dp"  
    android:text="No style!"/>
```

```
<TextView
```

```
    style="@style/RedButton"  
    android:text="style – RedButton!"/>
```

Selector

- Změna atributů View na základě změny stavu
 - `pressed`, `checked`, `selected` apod.
- `drawable`, `string`, `color`,...
- ale! -> View musí změnu stavu podporovat
- příklad!

Custom Shapes

- Definice pomocí XML
- Shape, Bitmap, Gradient apod.
- Příklad!

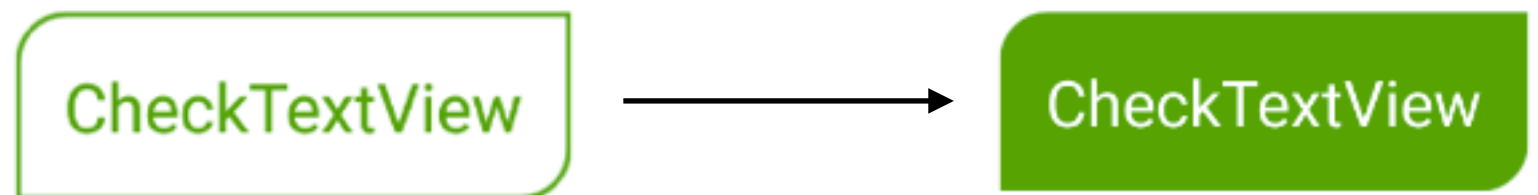
Animace

- `animateLayoutChanges = "true" <- animace zdarma`
- Value Animator
- `/res/anim`
 - frame-by-frame -> `<animation-list>`
 - matematicky definované -> `<scale>`
- příklad!

Cvičení

- RelativeLayout s potomky obsahující atributy:
 - above, toLeftOf, align_parent_bottom
 - měnit visibility těchto prvků s/bez animate layout changes
- Vytvořit CheckTextView:

`checkTextView.toggle()`



```
<CheckedTextView  
    android:id="@+id/checkTextView"  
    android:text="CheckedTextView"  
    style="@style/checkedViewStyle"/>
```