SeekBar (Слайдер)

**SeekBar** - это обычный слайдер, когда пользователь может передвигать ползунок пальцем на экране. Ползунок также можно двигать при помощи клавиш-стрелок.

Компонент **SeekBar** находится в разделе **Widgets** и наследуется от класса **ProgressBar**. В Android Studio 3.0 представлен в двух вариантах: **SeekBar** и **SeekBar (Discrete)**.

Для отслеживания перемещения ползунка **SeekBar** необходимо реализовать интерфейс **SeekBar.OnSeekBarChangeListener** с методами-заглушками:

* **onProgressChanged()** — уведомляет об изменении положения ползунка;
* **onStartTrackingTouch()** — уведомляет о том, что пользователь начал перемещать ползунок;
* **onStopTrackingTouch()** — уведомляет о том, что пользователь закончил перемещать ползунок

Основное событие компонента **SeekBar**, которое наиболее часто используется на практике - **SeekBar.OnSeekBarChangeListener**.

Создадим новый проект и добавим компоненты **SeekBar** и **TextView**:

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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="ru.alexanderklimov.as23.MainActivity">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="8dp"

android:layout\_marginRight="8dp"

android:textSize="24sp"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

tools:text="90"

app:layout\_constraintTop\_toTopOf="parent"

android:layout\_marginTop="8dp"

app:layout\_constraintHorizontal\_bias="0.501"/>

<SeekBar

android:id="@+id/seekBar"

android:layout\_width="368dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:max="100"

android:progress="3"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView"/>

</android.support.constraint.ConstraintLayout>

Напишем код, чтобы в текстовом поле отображалось текущее значение ползунка после того, как пользователь отпустит его.

package ru.alexanderklimov.seekbar;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.widget.SeekBar;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements SeekBar.OnSeekBarChangeListener {

private TextView mTextView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

final SeekBar seekBar = (SeekBar)findViewById(R.id.seekBar);

seekBar.setOnSeekBarChangeListener(this);

mTextView = (TextView)findViewById(R.id.textView);

mTextView.setText("0");

}

@Override

public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

}

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override

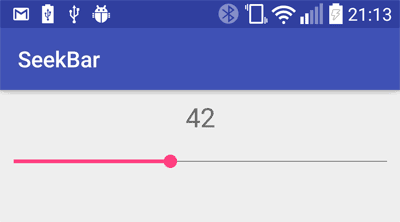
public void onStopTrackingTouch(SeekBar seekBar) {

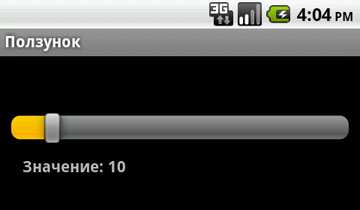
mTextView.setText(String.valueOf(seekBar.getProgress()));

}

}

Ниже представлены вариант на Android 5.0 и старый вариант на Android 2.3 до появления Material Design, чтобы вы видели, как менялся интерфейс.





Меняем задний фон экрана

Усложним пример и будем менять цвет у фона экрана. Добавим на форму три компонента **SeekBar**:

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.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:id="@+id/constrainLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="ru.alexanderklimov.as23.MainActivity">

<SeekBar

android:id="@+id/seekBar\_Red"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:max="255"

app:layout\_constraintBottom\_toTopOf="@+id/seekBar\_Green"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

/>

<SeekBar

android:id="@+id/seekBar\_Green"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:max="255"

app:layout\_constraintBottom\_toTopOf="@+id/seekBar\_Blue"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/seekBar\_Red"/>

<SeekBar

android:id="@+id/seekBar\_Blue"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:max="255"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/seekBar\_Green"/>

</android.support.constraint.ConstraintLayout>

Напишем код, меняющий значение цвета через значения ползунка:

package ru.alexanderklimov.seekbar;

import android.os.Bundle;

import android.support.constraint.ConstraintLayout;

import android.support.v7.app.AppCompatActivity;

import android.widget.SeekBar;

public class MainActivity extends AppCompatActivity {

private SeekBar mRedSeekBar, mGreenSeekBar, mBlueSeekBar;

private ConstraintLayout mScreenLayout;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mScreenLayout = (ConstraintLayout) findViewById(R.id.constrainLayout);

mRedSeekBar = (SeekBar) findViewById(R.id.seekBar\_Red);

mGreenSeekBar = (SeekBar) findViewById(R.id.seekBar\_Green);

mBlueSeekBar = (SeekBar) findViewById(R.id.seekBar\_Blue);

updateBackground();

mRedSeekBar.setOnSeekBarChangeListener(seekBarChangeListener);

mGreenSeekBar.setOnSeekBarChangeListener(seekBarChangeListener);

mBlueSeekBar.setOnSeekBarChangeListener(seekBarChangeListener);

}

private SeekBar.OnSeekBarChangeListener seekBarChangeListener = new SeekBar.OnSeekBarChangeListener() {

@Override

public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

updateBackground();

}

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override

public void onStopTrackingTouch(SeekBar seekBar) {

}

};

private void updateBackground() {

int redValue, greenValue, blueValue;

redValue = mRedSeekBar.getProgress();

greenValue = mGreenSeekBar.getProgress();

blueValue = mBlueSeekBar.getProgress();

// меняем фон через формат RGB

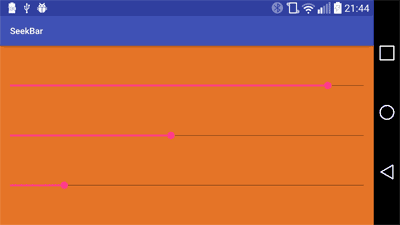
mScreenLayout.setBackgroundColor(0xff000000 + redValue \* 0x10000 + greenValue \* 0x100

+ blueValue);

}

}

Кот Рыжик попросил выставить цвет, наиболее подходящий цвету его шкурки. Я решил ему немного польстить:





Дополнительное чтение

[Продвинутые примеры для слайдера](http://developer.alexanderklimov.ru/android/views/seekbar-advanced.php)

[patryk1007/ShootingSlider](https://github.com/patryk1007/ShootingSlider) - прикольный слайдер, напоминающий стрельбу из пушки в играх

[bernaferrari/EmojiSlider: A slider widget rich in emoji and highly customisable.](https://github.com/bernaferrari/EmojiSlider) - слайдер с эмодзи-значками

[MarcinMoskala/ArcSeekBar: Good looking curved Android SeekBar](https://github.com/MarcinMoskala/ArcSeekBar)

[AnderWeb/discreteSeekBar](https://github.com/AnderWeb/discreteSeekBar) - интересная реализация

[neild001/SeekArc](https://github.com/neild001/SeekArc) - круговой ползунок. [Пост автора](http://www.inter-fuser.com/2013/11/seekarc-curvy-seekbar.html).

[Android Custom Seekbar with animation Example](http://iamvijayakumar.blogspot.ru/2014/04/android-custom-seekbar-with-animation.html) - ещё один пример кругового ползунка

[GoodieBag/ProtractorView: A semicircular seekbar view for selecting angle from 0° to 180°](https://github.com/GoodieBag/ProtractorView) - ползунок в виде полукруга.

[oli107/material-range-bar: Android widget for selecting a range of values.](https://github.com/oli107/material-range-bar) - не совсем ползунок, позволяет выбрать диапазон значений.

[lawloretienne/DiscreteSlider: A slider that allows a user to select a value at one of the specified tickmarks](https://github.com/lawloretienne/DiscreteSlider) - слайдер для выбора значений.

[akaita/CircularSeekBar: Simple custom Android View providing a Circular spin to SeekBars](https://github.com/akaita/CircularSeekBar)