

予報をレイアウトに表示する

WeatherForecast.javaのTemperatureクラスをリスト6-19のように変更します。

○リスト6-19: WeatherForecast.java

```
public static class Temperature {
    public final Value min;
    public final Value max;

    // 省略

+   @Override
+   public String toString() {
+       StringBuffer sb = new StringBuffer();

+       // 最低気温 / 最高気温
+       if (min.celsius != null) {
+           sb.append(min.celsius);
+       } else {
+           sb.append(" - ");
+       }
+       sb.append("°C / ");

+       if (max.celsius != null) {
+           sb.append(max.celsius);
+       } else {
+           sb.append(" - ");
+       }
+       sb.append("°C");

+       return sb.toString();
+   }

    public static class Value {
```

次にMainActivity.javaをリスト6-20のように変更します。

○リスト6-20: MainActivity.java

```
+ import android.view.View;
+ import android.widget.ImageView;
+ import android.widget.LinearLayout;

public class MainActivity extends AppCompatActivity {

-   private TextView result;
+   private TextView location; ①
+   private LinearLayout forecastsLayout; ②

    public class ApiTask extends GetWeatherForecastTask {
        @Override
        protected void onPostExecute(WeatherForecast data) {
            super.onPostExecute(data);
```

```
        if (data != null) {
            result.setText(data.location.area + " "
                + data.location.prefecture + " "
                + data.location.city);
+           location.setText(data.location.area + " "
+               + data.location.prefecture + " "
+               + data.location.city);

            for (WeatherForecast.Forecast forecast : data.forecastList) {
                // Windowsの場合は「¥」、MacやLinuxの場合は「/」を入力
                result.append("¥n");
                result.append(forecast.dateLabel + " " + forecast.telop);

+               View row =
+                   View.inflate(MainActivity.this, R.layout.forecast_row, null);
+               TextView date = (TextView) row.findViewById(R.id.tv_date);
+               date.setText(forecast.dateLabel);

+               TextView telop = (TextView) row.findViewById(R.id.tv_telop);
+               telop.setText(forecast.telop); ③

+               TextView temperature =
+                   (TextView) row.findViewById(R.id.tv_temperature);
+               temperature.setText(forecast.temperature.toString());

+               ImageView image = (ImageView) row.findViewById(R.id.iv_weather);

+               ImageLoaderTask task = new ImageLoaderTask();
+               task.execute(new ImageLoaderTask.Request(
+                   image, forecast.image.url)); ④

+               forecastsLayout.addView(row);
            }
        } else if (exception != null) {
            Toast.makeText(getApplicationContext(), exception.getMessage(),
                Toast.LENGTH_SHORT).show();
        }
    }

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

-       result = (TextView) findViewById(R.id.tv_result);
+       location = (TextView) findViewById(R.id.tv_location);
+       forecastsLayout = (LinearLayout) findViewById(R.id.ll_forecasts);

        new ApiTask().execute("400040");
    }
}
```

- ① TextViewのIDの変更に合わせて、フィールド名をresultからlocationに変更する
- ② 予報を表示するレイアウトforecastLayoutのフィールドを追加する
- ③ WeatherForecastに含まれる予報1件（1日分）につき、forecast_rowのレイアウトを生成して、表示内容を設定する
- ④ 予報画像はImageLoaderTaskで読み込み（ダウンロード）と表示を行う

実行 天気予報（予報日と内容）と天気画像を取得（ダウンロードして）表示します。



6-8 レイアウトの見栄えを調整する

前のステップでは個々の表示内容をレイアウトしましたが、文字や画像の大きさ、デザインのバランスはまったく変更していません。

このステップでレイアウトを調整して、表示する天気予報の見栄えを調整します。まず、activity_main.xmlをリスト6-21のように変更します。

○リスト6-21：activity_main.xml

```
<TextView
    android:id="@+id/tv_location"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
+   android:textSize="25sp" ①
    />

<LinearLayout
    android:id="@+id/ll_forecasts"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_below="@id/tv_location"
+   android:layout_marginTop="16dp" ②
    android:orientation="vertical">
</LinearLayout>
```

① 予報地点のテキストサイズを25spに設定する。spはScale-independent Pixelsの略で、表示するAndroidデバイスのディスプレイやユーザーのフォントサイズ設定に応じて大きさが自動調整される

② 上にマージンを16dpに設定する。dpはDensity-independent Pixelsの略で、表示するAndroidデバイスのディスプレイに応じて大きさ自動調整される。spと異なり、dpは自動調整の際にユーザーのフォントサイズ設定を考慮しない

次に、forecast_row.xmlをリスト6-22のように変更します。

○リスト6-22：forecast_row.xml

```
<TextView
    android:id="@+id/tv_date"
+   android:textSize="20sp" ①
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

<ImageView
    android:id="@+id/iv_weather"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
+   android:layout_margin="8dp"
+   android:minHeight="80dp"
+   android:minWidth="80dp" ②
    android:layout_toRightOf="@id/tv_date" />

<TextView
    android:id="@+id/tv_telop"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@id/iv_weather"
+   android:layout_marginBottom="8dp"
+   android:textSize="20sp" ③
    android:layout_toRightOf="@id/iv_weather" />

<TextView
    android:id="@+id/tv_temperature"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
+   android:textSize="20sp" ④
    android:layout_below="@id/tv_telop"
    android:layout_toRightOf="@id/iv_weather" />

</RelativeLayout>
```

- ① 予報日のテキストサイズを20spに設定する
- ② 予報画像のマージン（上下左右）をそれぞれ8dpに、最小サイズを縦横80dpに設定する
- ③ 予報のテキストサイズを20spに、下マージンを8dpに設定する
- ④ 気温のテキストサイズを16spに設定する