

メソッド一覧

本書の各 LESSON で利用した主だったメソッドをまとめました。

「サンプルとは違った使い方をしたい」など、ご自身の開発の幅を広げる際に、ご利用ください。

>> addAction(Notification.Action action)

参照P.374

クラス名 Notification.Builder

メソッドの役割 引数で指定した Notification.Action を通知アクションに追加します。

>> addItemDecoration(RecyclerView.ItemDecoration decor)

参照P.288

クラス名 RecyclerView

メソッドの役割 RecyclerView 上で表示しているリストアイテム間に視覚的にグループ分けできる描画を追加します。引数で指定している ItemDecoration は継承して自作することで任意の描画が可能になります。使用例としてよく区切り線に用いられます。

>> addOnPageChangeListener(ViewPager.OnPageChangeListener listener)

クラス名 ViewPager

メソッドの役割 引数で指定した listener をセットします。ページが切り替わった時のイベントを取得できます。

>> addRemoteInput(RemoteInput remoteInput)

参照P.373

クラス名 NotificationCompat.Action.Builder

メソッドの役割 通知にユーザー入力可能なアクションを追加します。引数には入力アクションの元となる remoteinput を指定します。

>> addToBackStack(String name)

参照P.189

クラス名 FragmentTransaction

メソッドの役割 Fragment の切り替わりをバックスタックに追加します。引数の name はどの時点のスタックかを示すためのタグとして使用します。

>> **addView(View child, ViewGroup.LayoutParams params)**

[参照P.134](#)

クラス名 ViewGroup

メソッドの役割 ViewGroup(LinearLayout や RelativeLayout) に View(TextView や EditText等)を追加します。第1引数は追加したいViewを第2引数にはviewのLayoutParamsを指定します。

>> **beginTransaction** メソッド

[参照P.325](#)

クラス名 Realm

メソッドの役割 Databaseへの書き込みトランザクションを開始します。Realmを使用してDatabaseへ書き込む際には必ずトランザクションを開始する必要があります。

>> **cancelTransaction** メソッド

[参照P.324](#)

クラス名 Realm

メソッドの役割 Databaseへの書き込みトランザクションを中止します。

>> **commitTransaction** メソッド

[参照P.325](#)

クラス名 Realm

メソッドの役割 Databaseへの書き込みトランザクションをコミットします。書き込み処理後に本メソッドを呼び出さないとDatabaseの書き込み処理は適用されません。

>> **createObject(Class<E> clazz)**

[参照P.325](#)

クラス名 Realm

メソッドの役割 引数で指定したクラスを元に新しく Realm オブジェクトを作成し、全フィールドをデフォルト値で埋めます。

>> **finish** メソッド

[参照P.154](#)

クラス名 Activity

メソッドの役割 Activityを終了させるメソッドです。任意のタイミングでActivityを終了したい時に呼び出します。

>> getActivity(Context context, int requestCode, Intent intent, int flags)

参照P.190

クラス名 PendingIntent

メソッドの役割 Activityを起動するためのPendingIntentオブジェクトを取得します。第2引数のrequestCodeにはどのPendingIntentからの指定かを識別するための数値を、第3引数のintentには起動するActivityクラスや付加データをセットします。

>> getBroadcast(Context context, int requestCode, Intent intent, int flags)

参照P.373

クラス名 PendingIntent

メソッドの役割 BroadcastReceiverを起動するためのPendingIntentオブジェクトを取得します。第2引数のrequestCodeにはどのPendingIntentからの指定かを識別するための数値を、第3引数のintentには起動するBroadcastReceiverクラスや付加データをセットします。

>> getCacheDir メソッド

参照P.308

クラス名 Context

メソッドの役割 内部キャッシュ領域を取得します。ファイルビューワー等を使ってユーザーアクセスすることができない領域です。あくまで一時保存のキャッシュとして使用します。

>> getDarkMutedColor(int defaultColor)

参照P.275

クラス名 Palette

メソッドの役割 画像データから色調を抑えた暗めの色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> getDarkMutedSwatch メソッド

参照P.276

クラス名 Palette

メソッドの役割 画像データから色調を抑えた暗めの色見本を取得します。

>> getDarkVibrantColor(int defaultColor)

参照P.275

クラス名 Palette

メソッドの役割 画像データから鮮やかな暗めの色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> getDarkVibrantSwatch メソッド

[参照P.276](#)**クラス名** Palette**メソッドの役割** 画像データから鮮やかな暗めの色見本を取得します。

>> getExternalCacheDir メソッド

[参照P.308](#)**クラス名** Context**メソッドの役割** 外部キャッシュ領域を取得します。ファイルビューワー等を使ってユーザーアクセスすることが可能な領域です。あくまで一時保存のキャッシュとして使用します。

>> getExternalFilesDir メソッド

[参照P.308](#)**クラス名** Context**メソッドの役割** 外部データ領域を取得します。ファイルビューワー等を使ってユーザーアクセスすることが可能な領域です。一定期間保持したいデータを保存する際に使用します。

>> getFilesDir メソッド

[参照P.308](#)**クラス名** Context**メソッドの役割** 内部データ領域を取得します。ファイルビューワー等を使ってユーザーアクセスすることができない領域です。一定期間保持したいデータを保存する際に使用します。

>> getLightMutedColor(int defaultColor)

[参照P.275](#)**クラス名** Palette**メソッドの役割** 画像データから色調を抑えた明るめの色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> getLightMutedSwatch メソッド

[参照P.276](#)**クラス名** Palette**メソッドの役割** 画像データから色調を抑えた明るめの色見本を取得します。

>> getLightVibrantColor(int defaultColor)

参照P.275

クラス名 Palette

メソッドの役割 画像データから鮮やかな明るめの色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> getLightVibrantSwatch メソッド

参照P.276

クラス名 Palette

メソッドの役割 画像データから鮮やかな明るめの色見本を取得します。

>> getMenuInfo メソッド

参照P.232

クラス名 MenuItem

メソッドの役割 ActivityやFragmentに追加しているメニューアイテムを取得します。

>> getMutedColor(int defaultColor)

参照P.275

クラス名 Palette

メソッドの役割 画像データから色調を抑えた色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> getMutedSwatch メソッド

参照P.276

クラス名 Palette

メソッドの役割 画像データから色調を抑えた色見本を取得します。

>> getResultsFromIntent (Intent intent)

参照P.371

クラス名 RemoteInput

メソッドの役割 RemoteInputからの結果を取得するためのIntentを指定します。例えば、通知上のボタンを押下した時に、このIntentに指定したActivityを呼び出すことができます。

>> `getService (Context context, int requestCode, Intent intent, int flags)`

[参照P.373](#)

クラス名 PendingIntent

メソッドの役割 Serviceを起動するためのPendingIntentオブジェクトを取得します。第2引数のrequestCodeにはどのPendingIntentからの指定かを識別するための数値を、第3引数のintentには起動するServiceクラスや付加データをセットします。

>> `getVibrantColor(int defaultColor)`

[参照P.275](#)

クラス名 getVibrantColor

メソッドの役割 画像データから鮮やかな色を取得します。取得できなかった場合は引数で指定したデフォルトカラーリソースを使用します。

>> `getVibrantSwatch` メソッド

[参照P.276](#)

クラス名 getVibrantColor

メソッドの役割 画像データから鮮やかな色見本を取得します。

>> `isInMultiWindowMode` メソッド

[参照P.365](#)

クラス名 Activity

メソッドの役割 Activityがマルチウィンドウモードかどうか調べる時に使用します。

>> `onActivityCreated (Bundle savedInstanceState)`

[参照P.190](#)

クラス名 Fragment

メソッドの役割 Activityが生成された(onCreate)後に呼ばれます。引数のsaveInstanceStateには以前データを保存していた場合、そのデータが格納された状態で呼び出されます。

» `onActivityResult(int requestCode, int resultCode, Intent data)`

参照P.154

クラス名 Activity

メソッドの役割 現在のActivity(アクティビティ A)から別のActivity(アクティビティ B)を呼び出し、そのActivity(アクティビティ B)が終了した後に呼ばれます。ただしstartActivityResultメソッドで呼び出したActivityしか感知できないので、このメソッドを利用したい時はstartActivityResultメソッドとセットで使います。第1引数はどのstartActivityResultから呼ばれたかを識別するためのrequestCodeを、第2引数には別のActivityで返されたresultCodeを、第3引数には別のActivityから渡されたdataが格納されています。

» `onAttach (Activity activity)`

参照P.190

クラス名 Fragment

メソッドの役割 Activityと関連付けられたタイミングで一度だけ呼ばれます。引数のactivityは関連付けられたactivityが格納されています。

» `onBindViewHolder (ViewHolder holder, int position)`

参照P.291

クラス名 RecyclerView

メソッドの役割 リストアイテムを表示するタイミングで呼ばれます。このメソッド内でリストアイテムの内容をセットします。第1引数はリストアイテムとなるViewHolderが、第2引数にはリストのポジションが格納されています。

» `onCreate(Bundle savedInstanceState)`

参照P.151

クラス名 Activity

メソッドの役割 Activity生成時に呼ばれます。このメソッド内でActivityのViewを生成したり初期化処理を行います。引数のsaveInstanceStateは通常nullが格納されていますが、onSaveInstanceStateをオーバーライドして値を保存すると、saveInstanceStateの値が更新されます。Activityのデータを一時保存しておくのに利用します。

» onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)

参照P.226

クラス名 Activity

メソッドの役割 コンテキストメニュー作成前に呼ばれます。このメソッド内でコンテキストメニューの構成を作成します。第1引数にはContextMenuのインスタンスが、第2引数にはメニューのビューのインスタンスが、第3引数にはメニュー情報が格納されています。

» onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)

参照P.291

クラス名 Fragment

メソッドの役割 Fragment生成時に呼ばれ、FragmentのViewを生成したり初期化処理を行います。saveInstanceStateには以前データを保存していた場合、そのデータが格納された状態で呼び出されます。

» onDataChange(DataSnapshot dataSnapshot)

参照P.348

クラス名 ValueEventListener

メソッドの役割 Firebase Databaseにて、データが変更された時にこのメソッドが呼ばれます。引数のdataSnapshotにはこのメソッドがコールバックで呼ばれた時点のDataSnapshotが格納されています。

» onDestroyView メソッド

参照P.190

クラス名 Fragment

メソッドの役割 Fragmentのビューが破棄された時に呼ばれます。

» onDestroy メソッド

参照P.322

クラス名 Activity

メソッドの役割 Activityが破棄された時に呼ばれます。

» onDetach メソッド

参照P.190

クラス名 Fragment

メソッドの役割 Fragmentが、Activityから取り外された時に呼ばれます。

>> onMultiWindowModeChanged (boolean isInMultiWindowMode)

参照P.366

クラス名 Activity

メソッドの役割 Android Nougat以降動作するメソッドで、マルチウィンドウにした時や、マルチウィンドウから元に戻した時に呼ばれます。引数のisInMultiWindowModeがtrueの場合は現在マルチウィンドウモード、falseの場合は通常ウィンドウモードと判断することができます。

>> onNewIntent (Intent intent)

参照P.158

クラス名 Activity

メソッドの役割 Activityが起動している間、再度Activityが起動した時に呼ばれます。AndroidManifestにて、launchModeがsingleTopなどに指定されている場合に呼ばれます。

>> onPauseメソッド

参照P.152

クラス名 Activity

メソッドの役割 Activityがバックグラウンドに移った場合に呼ばれます。

>> onReceive (Context context, Intent intent)

参照P.177

クラス名 Broadcast

メソッドの役割 Broadcastメッセージを受け取った時に呼ばれます。第2引数のintentにはBroadcastメッセージを送信した時に設定したIntentデータが含まれています。

>> onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults)

参照P.311

クラス名 Activity

メソッドの役割 パーミッションの許可ダイアログの結果を受け取るメソッドです。第1引数にはパーミッション要求を実施した際のリクエストコード、第2引数には要求したパーミッション群が、第3引数にはパーミッションの要求結果が格納されています。

≫ onRestart メソッド

[参照P.152](#)**クラス名** Activity**メソッドの役割** 停止していた Activity を再度開始する時に呼ばれるメソッドです。

≫ onRestoreInstanceState (Bundle savedInstanceState)

[参照P.157](#)**クラス名** Activity**メソッドの役割** onSaveInstanceState メソッドで保存した値を復元するメソッドです。引数の savedInstanceState に値が入っています。

≫ onResume メソッド

[参照P.152](#)**クラス名** Activity**メソッドの役割** Activity がフォアグラウンドに移った場合に呼ばれます。

≫ onSaveInstanceState (Bundle outState)

[参照P.157](#)**クラス名** Activity**メソッドの役割** Activity が停止する前に呼ばれます。引数の outState オブジェクトに、現在の状態を保持するために必要な情報を入れておき、再起動したときに onRestoreInstanceState メソッドから取り出すことができます。

≫ onStartCommand (Intent intent, int flags, int startId)

[参照P.166](#)**クラス名** Service**メソッドの役割** サービスが起動した時や、再起動した時に呼ばれます。多くの場合、startService メソッドで起動した際に呼ばれます。Intent は null になることがあるので注意してください。

≫ onStart メソッド

[参照P.362](#)**クラス名** Activity**メソッドの役割** Activity が表示される前に呼ばれます。

≫ onStop メソッド

[参照P.152](#)**クラス名** Activity, Service**メソッドの役割** Activity が停止した時に呼ばれます。

» onUnbind (Intent intent)

参照P.166

クラス名 Service**メソッドの役割** サービスとの接続が解除された時に呼ばれます。

» onWindowFocusChanged (boolean hasFocus)

参照P.216

クラス名 Activity

メソッドの役割 画面からフォーカスが外れたり、逆にフォーカスしたりした時に呼ばれます。ライフサイクル的には、onResumeメソッドの後や、onPauseメソッドの前に呼ばれます。引数のhasFocusには現在のウィンドウのフォーカス状況が格納されています。trueの場合はフォーカス、falseの場合はフォーカスされていないことを示します。

» popBackStack (String name, int flags)

参照P.225

クラス名 Fragment

メソッドの役割 重なっているFragmentを取り除きます。引数にnullを指定すると、一番上のFragmentを取り除きます。これは非同期で実行されます。このメソッドは、あらかじめFragmentManagerのaddToBackStackメソッドを指定した時に使用します。第1引数には戻りたいトランザクションのname (addToBackStackメソッドで指定したもの)を、第2引数のflagsには0(そのトランザクションまで戻る)かFragmentManager.POP_BACK_STACK_INCLUSIVE(そのトランザクションを追加する前まで戻る)を指定します。

» post (Runnable r)

参照P.200

クラス名 Handler

メソッドの役割 引数に指定したRunnableインターフェースのインスタンスを、Handlerのキューに詰めます。Runnableのrunメソッドが実行されるのは、Handlerに設定されたLooper上のスレッドで実行されます。通常このスレッドはメインスレッドです。

» registerReceiver (BroadcastReceiver receiver, IntentFilter filter)

参照P.180

クラス名 Context

メソッドの役割 Activityにブロードキャストレシーバを登録します。第1引数は登録するBroadcastReceiverを、第2引数にはBroadcastReceiverのトリガーやフィルタを登録するためのIntentFilterを指定します。

» requestPermissions(Activity activity, String[] permissions, int requestCode)

参照P.310

クラス名 ActivityCompat**メソッドの役割** 引数に指定したパーミッションに対する許可を、ユーザーに求めるダイアログを表示します。

» runOnUiThread(Runnable action)

参照P.336

クラス名 Activity**メソッドの役割** 別スレッド上でこのメソッドを呼び出すと、引数に指定したRunnableインターフェースのrunメソッドが、メインスレッド(UIスレッド)上で呼ばれます。

» sendBroadcast(Intent intent)

参照P.178

クラス名 Activity, LocalBroadcastManager**メソッドの役割** 引数に指定したIntentの情報をもとに、BroadcastMessageを送信します。受け取るには、registerBroadcastメソッド等を使用し設定しておく必要があります。

» sendMessage(Message msg)

参照P.201

クラス名 Handler**メソッドの役割** 引数に指定したメッセージを送信します。メッセージはhandleMessageメソッドをオーバーライドすることで受け取ることができます。

» setArguments(Bundle args)

参照P.191

クラス名 Fragment**メソッドの役割** Fragmentに引数を設定します。このメソッドはFragmentTransactionのcommitメソッドを呼ぶ前に設定します。設定した値はgetArgumentsメソッドで取得できます。

» setDefaultConfiguration(RealmConfiguration configuration)

参照P.322

クラス名 Realm**メソッドの役割** アプリケーションの"files"ディレクトリにRealm ファイルを作成するRealmConfiguration(引数で指定)を作成します。

>> setFlags(int flags)

参照P.157

クラス名 Intent

メソッドの役割 Intentを送る対象に対して、特定の命令をする時に使用します。このメソッドを使用する前に、既にflagsを指定している場合、上書きされるので注意しましょう。上書きしたくない場合は、getFlagsメソッドを組み合わせるか、addFlagsメソッドを使用します。

>> setGroup(String groupKey)

参照P.381

クラス名 NotificationCompat.Builder

メソッドの役割 複数の通知をまとめます。引数で指定したgroupKeyが同じグループであるかどうかを示すものとなります。

>> setGroupSummary(boolean isGroupSummary)

参照P.381

クラス名 NotificationCompat.Builder

メソッドの役割 通知の詳細を有効にするかを指定します。setGroupメソッドとセットで使用し、引数のisGroupSummaryは多くの場合trueを指定します。

>> setHeaderIcon(int iconRes)

参照P.232

クラス名 ContextMenu

メソッドの役割 引数で指定したアイコンリソースをContextMenuのヘッダーアイコンに設定します。

>> setHeaderTitle(CharSequence title)

参照P.231

クラス名 ContextMenu

メソッドの役割 引数で指定したアイコンリソースをContextMenuのヘッダータイトルに設定します。

>> setLabel(CharSequence label)

参照P.373

クラス名 RemoteInput.Builder

メソッドの役割 引数のlabelを用いてDirectReplyの入力欄に表示される文字を定義します。

>> **setItemClickListener(AdapterView.OnItemClickListener listener)**

[参照P.287](#)

クラス名 ListView, GridView

メソッドの役割 引数で指定したListenerをセットします。アイテムをタップした時のイベントを取得できます。

>> **setOnTabChangeListener(TabHost.OnTabChangeListener l)**

[参照P.124](#)

クラス名 TabHost

メソッドの役割 引数で指定したListenerをセットします。タブが切り替わった時のイベントを取得できます。

>> **setRadius(float radius)**

[参照P.260](#)

クラス名 CardView

メソッドの役割 カードの角の丸みを指定します。いわゆる角丸を作成するためのメソッドで引数にはどのくらいの丸みにするかの数値を指定します。

>> **setRemoteInputHistory(CharSequence[] text)**

[参照P.376](#)

クラス名 NotificationCompat.Builder

メソッドの役割 通常の通知の下に、小さな通知を追加します。例えば、チャット履歴を表示することができます。引数のtextはCharSequenceの配列になっていて、これらの配列が全て履歴として表示されます。

>> **setResult (int resultCode)**

[参照P.154](#)

クラス名 Activity

メソッドの役割 Activity 終了時の resultCode を指定します。resultCode は、呼び元の onActivityResult メソッドの引数に渡されます。

>> **setStream(InputStream bitmapData)**

[参照P.241](#)

クラス名 WallpaperManager

メソッドの役割 引数に指定した画像データのストリームを壁紙として設定します。画像データのストリーム化には、FileInputStream 等を使用します。

>> sort(String fieldName)

参照P.324

クラス名 Realm**メソッドの役割** 引数で指定したフィールド名で検索結果を昇順ソートします。

>> startActivity (Intent intent)

参照P.141

クラス名 Context, Activity**メソッドの役割** Intentに指定したActivityを開始します。

>> startActivityForResult (Intent intent, int requestCode)

参照P.153

クラス名 Activity**メソッドの役割** 基本的にはstartActivityメソッドと同じですが、開始したActivityが終了された時、onActivityResultメソッドで結果を受け取ることができます。どのActivityが対象かを判断するために引数でrequestCodeを指定します。

>> startService(Intent service)

参照P.165

クラス名 Context**メソッドの役割** Intentに指定したServiceを開始します。

>> unregisterReceiver(BroadcastReceiver receiver)

参照P.180

クラス名 Context**メソッドの役割** registerReceiverメソッドで登録したメソッドを登録解除します。引数のreceiverには登録時と同じBroadcastReceiverを指定します。

>> updateChildren(Map<String, Object> update)

参照P.350

クラス名 DatabaseReference**メソッドの役割** 引数で指定した値で、データを更新します。

>> where(Class<E> clazz)

参照P.324

クラス名 Realm**メソッドの役割** 引数に指定したクラスを中心にデータを検索します。このメソッドの後に続けて条件を記述します。