



## Android入門 アプリ開発の基本、ビューのライフサイクル(図解)



Android アプリケーション開発では、いつどのイベントが発生するかを知っておくというのは大事な事です。

アクティビティを実装する場合はアクティビティのイベントがいつ発生するか(ライフサイクル)が大事ですし、自分でビューを作る時はビューのイベントを把握しておくともスムーズに実装ができるようになります。

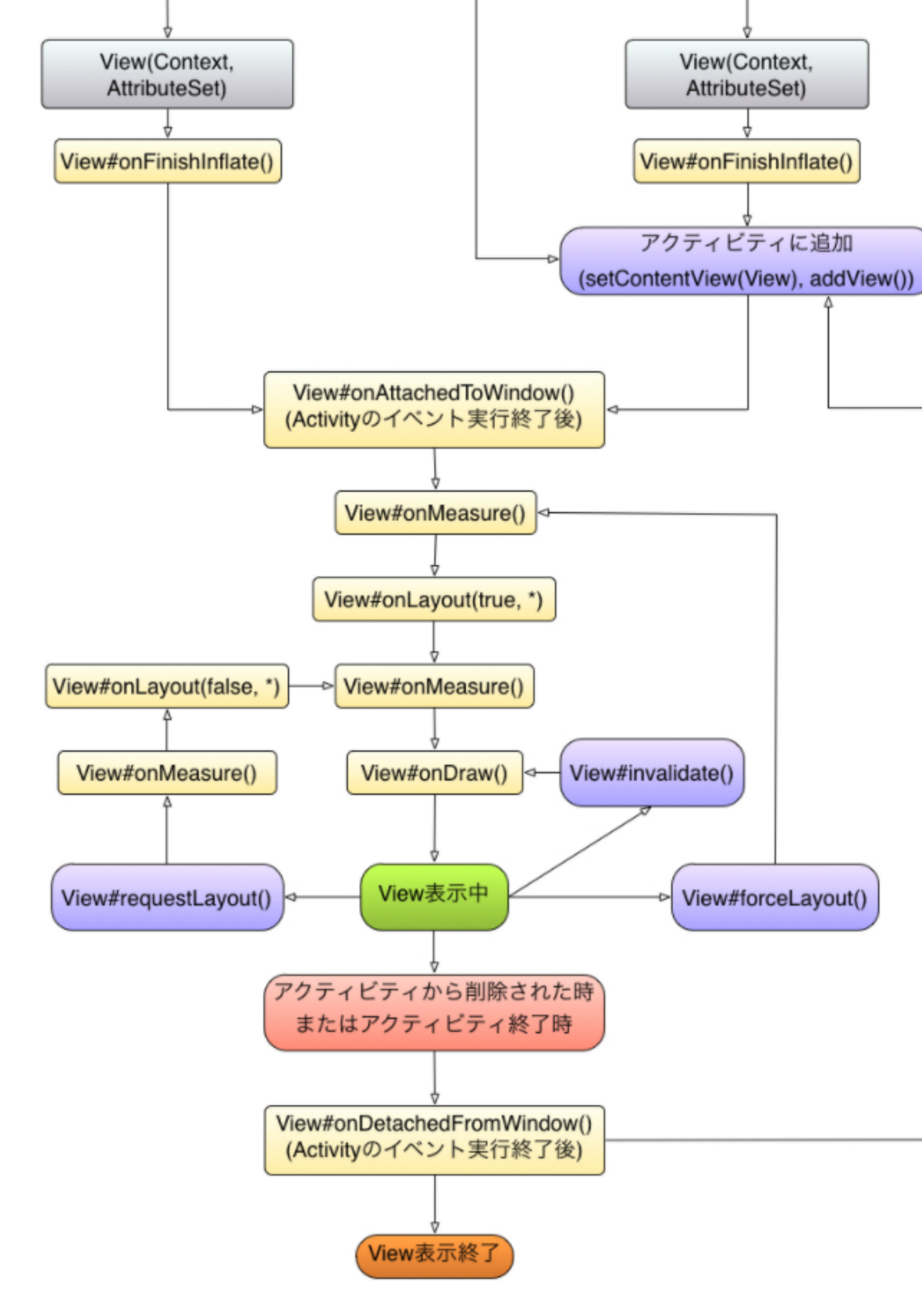
Androidのアクティビティについてはいつどのイベント公式サイトのアクティビティライフサイクルの項目で分かりやすく図示されているのですが、ビューのライフサイクル、イベントの詳細を見つけれなかったため、今回は整理も兼ねてこの記事を公開することにしました。

Android の View は Activity#setContentView() を呼ぶだけでも onFinishInflate(), onAttachedToWindow(),onLayout(),onDraw(),onMeasure()といったイベントが発生するのをご存知でしょうか？

どのタイミングでどのイベントが呼ばれるのか？

カスタムビューを作る時に処理をどのイベントに実装すべきかを悩んでいる方の助けになれば幸いです。

図:Viewのライフサイクル(View lifecycle: English version)



### 基本

ビューをアクティビティに割り当てる方法は大きく分けて3通りあります。

- **setContentView()**を使ったXMLリソースの静的割り当て  
Activity#setContentView(int:レイアウトリソースID)
- **inflate()** を使ったXMLリソースの静的割り当て  
Inflater#inflate(int:レイアウトリソースID), setContentView()
- 自分でインスタンス化してからの動的割り当て  
Inflater#inflate(int:レイアウトリソースID), setContentView()

基本的にビューのライフサイクルはアクティビティのライフサイクルにほとんど依存しません。

唯一アクティビティの終了する時だけは onDetachedFromWindow()が実行されますが、

その他の画面の切り替えなどの影響を受けることはないようです。

### イベント一覧

#### View#onFinishInflate()イベント

名前の通り、XMLから展開された直後に呼ばれます。

#### View#onAttachedToWindow()イベント

ウィンドウへの割り当てが終わった後に非同期で呼ばれます。

特に、アクティビティのライフサイクルに関わるイベント (onCreate(),onResume(),onRestart()など)は、すべて実行が終わってから onAttachedToWindow()が実行されるようです。



#### View#onMeasure()イベント

ビューのサイズを決定するイベントです。

ビューがウィンドウに割り当てられている間、頻繁に実行されます。

このメソッドはサイズ計算、設定に必要な処理以外を加えるべきではありません。

#### View#onLayout()イベント

子要素を持つビューなどでレイアウトを更新するためのイベントで、子ビューの情報や構成を更新する場合などに使用できます。

requestLayout()やforceLayout()が呼ばれた後に非同期で発生します。画面サイズが変わった場合などにもrequestLayout()やforceLayout()を通じて間接的に呼ばれます。

第一引数は画面に変更があったかどうかで、forceLayout()を呼ばれた場合やウィンドウ割り当て時はtrueが入り、requestLayout()を呼ばれた場合はfalseが入ります。

falseでの実行はtrueよりも多くあるので、負荷にコストがかかり頻繁に実行不要なものはtrueのみ実行する、という判定を加えるとよいでしょう。

#### View#onDraw()イベント

画面の描画を行うイベントです。Canvasを受け取って自身の描画処理(図形や画像の描画)を行うもので、一般的には子ビューを持たないもので使用されるイベントです。

初回割り当て時やrequestLayout(), forceLayout()で発生し、onLayout()の後に実行されます。

また、onLayout()と異なりinvalidate()でも発生するイベントです。

このメソッドは頻繁に呼ばれるため、負荷がかかる処理には不向きです。onDraw()で遅延が発生するとビューの描画が遅れることになります。

#### View#onDetachedFromWindow()イベント

アクティビティから割り当てがなくなった時に発生するイベントです。

アクティビティの終了時や、setContentView()で他のビューを割り当てられた時、親のビューに対してremoveAllViews()が呼ばれた時に発生します。

ライフサイクルに関係するイベントは以上です。他にもビューサイズの変更で発生するイベント、フォーカスや表示切り替えで発生するイベントなどもありますが、表示が複雑になるため省略しています。

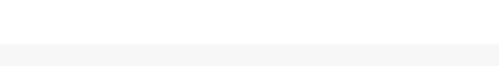
その他のイベントについては公式リファレンスが参考になります。

ご覧頂いた通り Androidではビューのイベントはアクティビティのイベントにはあまり関係しません。

このため、たとえば、ActivityのonResume(),onPause()などに合わせた処理を入れたい場合は明示的に開始、中断のメソッドを提供する必要があります。

検証用プロジェクトソースとアプリapkファイル:

アプリ本体(apk) プロジェクトソース(zip)



前

【Android】 ndk-build clean でシンタックスエラー。原因はNDK r8eのバグか？

次ページへ

Google App Engine でアプリケーション名を取得するには？