

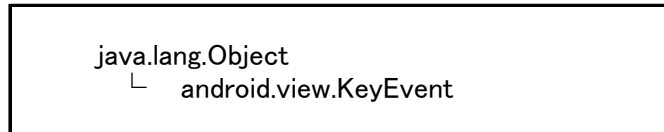
## キーイベント

- 2-1 KeyEvent概要
- 2-2 KeyEvent.Callback概要
- 2-3 アクティビティでキーイベントに応答
- 2-4 ビューでキーイベントに応答
- 2-5 アクティビティの「dispatchKeyEvent」メソッド
- 2-6 キーコード定数

## 2-1 KeyEvent概要

KeyEventクラスはアンドロイド端末のキーに対するイベントの定数、基本メソッドを提供するオブジェクトです。

クラス図は次のようになっています。



## 2-2 KeyEvent.Callback概要

KeyEvent.Callbackインターフェースはキーイベントの基底コールバック関数を提供しています。  
KeyEvent.Callbackインターフェースを継承するクラスには、それぞれのコールバック関数を実装してあれば、対応するキーイベントに応答が実現出来ます。

アクティビティ、ビュークラスはKeyEvent.Callbackインターフェースを継承しています。

キーイベントのコールバック関数は以下のように三つ用意しています。

メソッド	概要
boolean onKeyDown(int keyCode, KeyEvent event)	キーを押した瞬間に呼ばれるメソッド。
boolean onKeyLongPress(int keyCode, KeyEvent event)	キーを押し続いたとき呼ばれるメソッド。
boolean onKeyUp(int keyCode, KeyEvent event)	キーを押してからリリースしたとき呼ばれるメソッド。

## 2-3 アクティビティでキーイベントに応答

### 2-3-1 「onKeyDown」、「onKeyUp」メソッド

キーダウン(或はキーアップ)イベントが発生する時、アクティビティに「onKeyDown」(或は「onKeyUp」)メソッドを呼び出します。渡される引数「keyCode」により、押されたキーに特定する対処の実装が出来ます。

#### サンプルコード

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    switch (event.getKeyCode()) {  
        case KeyEvent.KEYCODE_DPAD_LEFT: // 十字左方キー  
            showToast(this, "左キー(onKeyDown)");  
            break;  
        case KeyEvent.KEYCODE_DPAD_CENTER:  
            break;  
        default:  
            break;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

※「onKeyUp」メソッドの実装方法は同じなので省略します。

## 実行例

上記ソースで実行し、左方キーを押すとトーストが現れます。



### 2-3-2 「onKeyLongPress」メソッド API level 5 以上

キーを押し続けるイベントが発生する時、アクティビティに「onKeyLongPress」メソッドを呼び出します。  
渡される引数「keyCode」により、押されたキーに特定する対処の実装が出来ます。

注意点として、「onKeyLongPress」コールバックを利用するとき、「onKeyDown」コールバックの実装を以下のようになさなければなりません。

- ・イベントオブジェクトの「startTracking」メソッドを呼びます。
- ・戻り値を「true」に設定します。

#### サンプルコード

```
public boolean onKeyLongPress(int keyCode, KeyEvent event) {  
    switch (keyCode) {  
        case KeyEvent.KEYCODE_DPAD_CENTER: // 十字中央キー  
            TestKeyEvent.showToast(TestKeyEvent.this, "中央キー(onKeyLongPress)");  
            break;  
        default:  
            break;  
    }  
    return super.onKeyLongPress(keyCode, event);  
}  
  
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    event.startTracking();  
    return true;  
}
```

## 実行例

上記ソースで実行し、中央キーを押し続けるとトーストが現れます。



## 2-4 ビューでキーイベントに応答

アクティビティと同じ、ビュークラスはKeyEvent.CallBackインターフェースを継承しますので、ビュークラスのキーイベントコールバック関数を実装してあれば、キーイベントに応答が出来ます。

### 2-4-1 「onKeyDown」、「onKeyUp」メソッド

キーダウンイベント(或はキーアップ)が発生する時、「onKeyDown」メソッド(或い「onKeyUp」)を呼び出します。渡される引数「keyCode」により、押されたキーに特定する対処の実装が出来ます。

#### サンプルコード

```
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    switch (event.getKeyCode()) {  
        case KeyEvent.KEYCODE_DPAD_LEFT: // 十字左方キー  
            showToast(this, "左キー(onKeyDown)");  
            break;  
        case KeyEvent.KEYCODE_DPAD_CENTER:  
            break;  
        default:  
            break;  
    }  
    return super.onKeyDown(keyCode, event);  
}
```

※「onKeyUp」メソッドの実装方法は同じなので省略します。

## 実行例

上記ソースで実行し、左方キーを押すとトーストが現れます。





キーを押し続けるイベントが発生する時、ビューの「onKeyLongPress」メソッドを呼び出します。  
渡される引数「keyCode」により、押されたのキーに対処する実装が出来ます。

アクティビティと同じ、「onKeyLongPress」コールバックを利用するとき、「onKeyDown」コールバックの実装を以下のようにしなければなりません。

- ・イベントオブジェクトの「startTracking」メソッドを呼びます。
- ・戻り値を「true」に設定します。

#### サンプルコード

```
public boolean onKeyLongPress(int keyCode, KeyEvent event) {  
    switch (keyCode) {  
        case KeyEvent.KEYCODE_DPAD_CENTER: // 十字中央キー  
            TestKeyEvent.showToast(TestKeyEvent.this, "中央キー(onKeyLongPress)");  
            break;  
        default:  
            break;  
    }  
    return super.onKeyLongPress(keyCode, event);  
}  
  
public boolean onKeyDown(int keyCode, KeyEvent event) {  
    event.startTracking();  
    return true;  
}
```

## 実行例

上記ソースで実行し、中央キーを押し続けるとトーストが現れます。



## 2-5 アクティビティの「dispatchKeyEvent」メソッド

KeyEvent.CallBackインターフェースの以外、アクティビティクラスはWindow.CallBackインターフェースも継承しています。Window.CallBackインターフェースでは「dispatchKeyEvent」コールバック関数を提供しますので、それを実装してあればキーイベントに応答することも出来ます。

「dispatchKeyEvent」メソッドを実装するとき、引数であるキーイベントの「getAction」、「getKeyCode」メソッドを利用すると、どのキーイベント及びどのキーの動作を把握でき、それに特定する対処の実装が出来ます。

本サンプルに使われるキーイベントのメソッド

メソッド	概要
final int getAction()	該当キーイベントの動作をゲットする。(動作: ACTION_DOWN、ACTION_UP※)
final int getKeyCode()	該当キーイベントのキーコードをゲットする。

サンプルコード

```
public boolean dispatchKeyEvent(KeyEvent event) {
    switch(event.getAction()) {
        case KeyEvent.ACTION_UP:
            switch (event.getKeyCode()) {
                case KeyEvent.KEYCODE_DPAD_CENTER: // 十字中央キー
                    showToast(this,"中央キー");
                    break;
                default:
                    break;
            }
            break;

        case KeyEvent.ACTION_DOWN:
            break;
        default:
            break;
    }
    return super.dispatchKeyEvent(event);
}
```














※ACTION\_DOWN: キーが押さえつけるとき発生  
ACTION\_UP: キーが押してからリリースするとき発生

## 実行例

上記ソースで実行し、中央キーを押すとトーストが現れます。



## 2-6 キーコード定数

KEYCODE_0	
KEYCODE_A	
KEYCODE_ALT_LEFT	
KEYCODE_ALT_RIGHT	
KEYCODE_AT	
KEYCODE_BACK	
KEYCODE_CALL	
KEYCODE_CAMERA	
KEYCODE_COMMA	
KEYCODE_DEL	
KEYCODE_DPAD_CENTER	
KEYCODE_DPAD_DOWN	
KEYCODE_DPAD_LEFT	

KEYCODE\_DPAD\_RIGHT



KEYCODE\_DPAD\_UP



KEYCODE\_ENDCALL



KEYCODE\_ENTER



KEYCODE\_HOME



KEYCODE\_MENU



KEYCODE\_PERIOD



KEYCODE\_POWER



KEYCODE\_SEARCH



KEYCODE\_SHIFT\_LEFT



KEYCODE\_SLASH



KEYCODE\_SPACE



KEYCODE\_SYM



KEYCODE\_VOLUME\_DOWN



KEYCODE\_VOLUME\_UP



※数字キー0から9までのコードは「KEYCODE\_0」から「KEYCODE\_9」までです。  
英文字キーAからZまでのコードは「KEYCODE\_A」から「KEYCODE\_Z」までです。