



Trang chủ

- [PHP](#)
- [Liên hệ](#)



- [HTML](#)
- [CSS](#)
- [JavaScript](#)
- [PHP](#)
- [Zend Framework](#)
- [Server](#)
- [Java](#)
- [Android](#)

- [Liên hệ](#)

Lập trình Android - Java §1) [Android Studio - Ứng dụng đầu tiên](#) §2) [Thành phần ứng dụng](#) §3) [Vòng đời Activity](#) §4) [Activity cơ bản](#) §5) [Lưu và phục hồi trạng thái](#) §6) [View, ViewGroup, Layout](#) §7) [TextView](#) §8) [Button](#) §9) [CheckBox](#) §10) [Switch / SwitchCompat](#) §11) [ToggleButton](#) §12) [RadioButton](#) §13) [TextClock](#) §14) [EditText](#) §15) [ImageView / ImageButton](#) §16) [FrameLayout](#) §17) [LinearLayout](#) §18) [ListView](#) §19) [ConstraintLayout](#) §20) [TableLayout](#) §21) [RelativeLayout](#) §22) [Drawable - shape](#) §23) [BitmapDrawable](#) §24) [VectorDrawable](#) §25) [NinePatchDrawable](#) §26) [LayerDrawable](#) §27) [StateListDrawable](#) §28) [ActionBar - Toolbar](#) §29) [Snackbar](#) §30) [FAB - FloatingActionButton](#) §31) [RecyclerView](#) §32) [CoordinatorLayout 1](#) §33) [CoordinatorLayout 2](#) §34) [Broadcast Intent](#) §35) [Theme Android \(1\)](#) §36) [Theme Android \(2\)](#) §37) [SQLite \(1\)](#) §38) [SQLite \(2\)](#) §39) [Màn hình Ngang - Đọc](#)

- [Lập trình PHP](#)
 - [PHP Cơ bản](#)
 - [SPL](#)
 - [PSR](#)
- [Lập trình ứng dụng iOS - Swift](#)
- [Lập trình Dart - Flutter](#)
- [Lập trình C# \(C Sharp\)](#)
- [Xenforo](#)
- [Zend Framework](#)
 - [Expressive](#)
- [Server](#)
 - [MySql Server](#)
 - [Apache](#)
 - [PHP](#)
- [HTML](#)
- [Javascript](#)
 - [JQuery](#)
 - [TypeScript - Angular](#)
- [CSS](#)
 - [Sử dụng SASS / SCSS](#)
 - [Bootstrap - CSS Framework](#)
- [SQL](#)
 - [SQL Server \(.NET Framework - C#\)](#)
- [Java](#)
 - [Android Java](#)
- [Thuật ngữ - Các vấn đề cơ bản](#)
- [Tools](#)
 - [Git và GitHub](#)
 - [Grunt](#)
 - [Docker](#)
 - [macOS](#)
 - [English Study](#)
- [Tin tức công nghệ](#)
- [Java](#)
- [Android Java](#)

Lưu và phục hồi trạng thái của Activity trong Android


Tìm hiểu và thực hành cách thức lưu và phục hồi trạng thái với các phương thức `onSaveInstanceState`, `onRestoreInstanceState` cũng như đề cập lớp `Bundle`


Phần này tạo một ứng dụng đơn giản tên là `SaveAndRestoreState` để tìm hiểu việc lưu lại trạng thái của Activity và phục hồi trạng thái của Activity trong vòng đời hoạt động của nó.



Tạo Project - SaveAndRestoreState

Mở **Android Studio** chọn tạo Project mới, nhập tên ứng dụng là **SaveAndRestoreState**

 Create New Project

 Create Android Project

Application name

Company domain

Project location

 ...


Package name

net.xuanthulab.saveandrestorestate

Edit

☐ Include C++ support

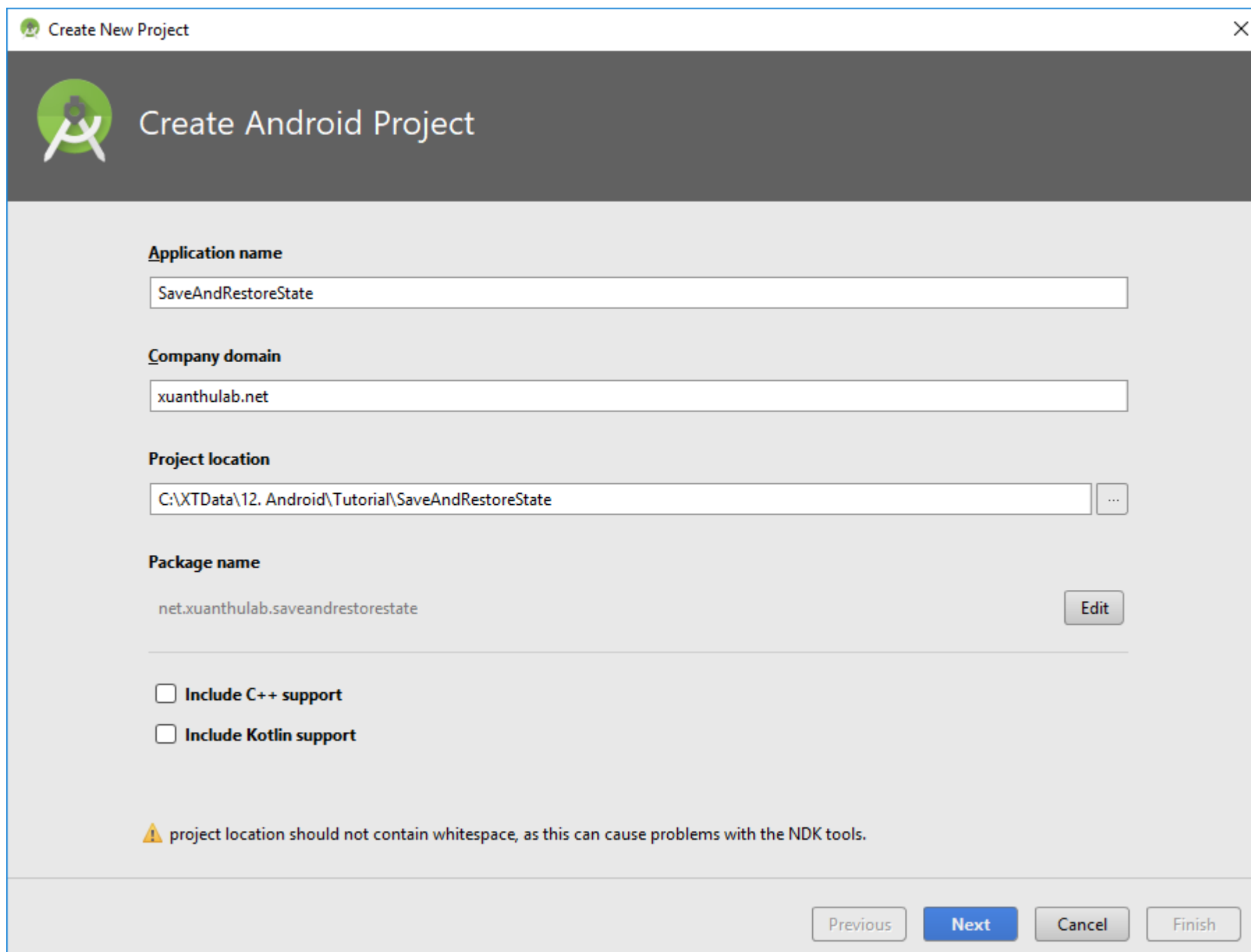
☐ Include Kotlin support

 project location should not contain whitespace, as this can cause problems with the NDK tools.

Previous Next Cancel Finish

Trong cửa sổ tiếp theo chọn **Phone and Table** và API mong muốn. Bấm tiếp, của sổ **Add an Activity to Mobile** chọn **Empty Activity**

Trong cửa sổ **Configure Activity** điền tên Activity là: **SaveRestoreActivity** và layout tên là **activity_save_restore** như hình dưới. Sau đó Bấm vào Finish



Create New Project

Create Android Project

Application name

SaveAndRestoreState

Company domain

xuanthulab.net

Project location

C:\XTData\12. Android\Tutorial\SaveAndRestoreState

Package name

net.xuanthulab.saveandrestorestate

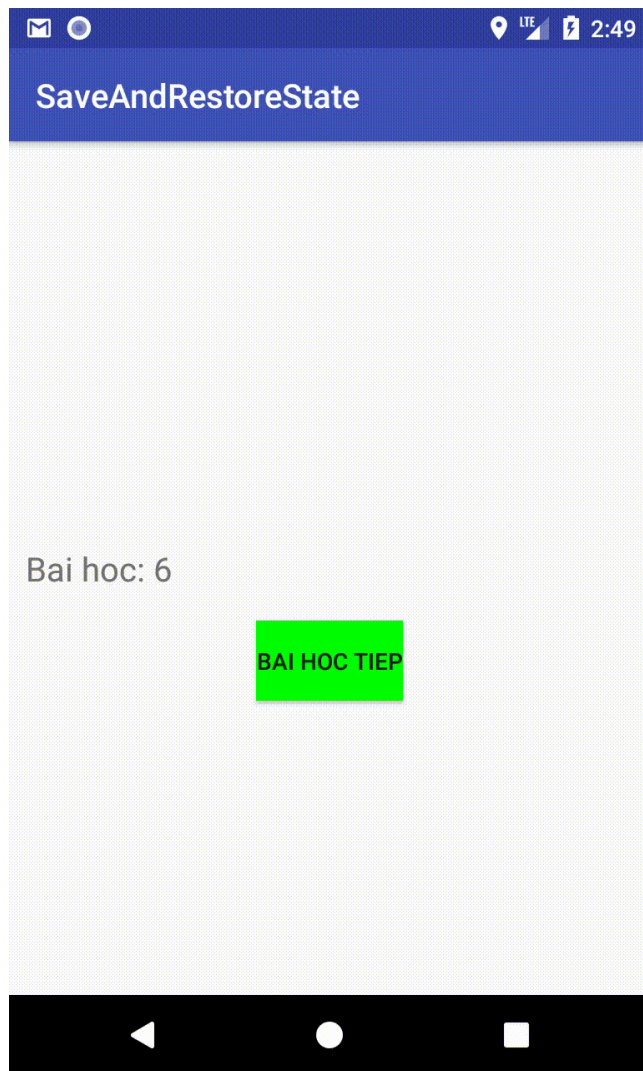
☐ Include C++ support

☐ Include Kotlin support

⚠ project location should not contain whitespace, as this can cause problems with the NDK tools.

Previous Next Cancel Finish

Ta sẽ xây dựng **SaveRestoreActivity** hiển thị một TextView chứa tên bài học, một nút bấm mà mỗi khi bấm vào thì tên bài học thay đổi. Khi nút bấm được bấm vào thì màu nền thay đổi từ màu ban đầu sang màu `Color.GREEN` như kết quả sau:



Cập nhật lại layout `activity_save_restore.xml` thêm TextView và Button với ID tương ứng là `message` và `nextlesson`

```
<?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="net.xuanthulab.saveandrestorestate.SaveRestoreActivity">
```

<TextView

```
    android:id="@+id/message"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Bai hoc so: ?"
    android:padding="10dp"
    android:textSize="20sp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
    android:id="@+id/next_lesson"

    android:background="@android:color/holo_blue_dark"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Bai hoc tiep"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/message" />
```

```
</android.support.constraint.ConstraintLayout>
```

Tiếp theo cập nhật code Java cho file: SaveRestoreActivity.java như sau:

```
public class SaveRestoreActivity extends AppCompatActivity {

    private int mLesson = 1; //Lưu tên bài hiện tại
    TextView message;
    Button button;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_save_restore);

        //Lấy đối tượng TextView và Button trong layout
        message = findViewById(R.id.message);
        button = findViewById(R.id.nextlesson);
    }
}
```

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        //bấm vào thì bài học tăng 1 và đặt  
        //màu nền của nút bấm thành GREEN,  
        mLesson++;  
        setLesson();  
        button.setBackgroundColor(Color.GREEN);  
    }  
});  
  
setLesson();  
  
}  
  
//Hiện thị dòng chữ tên bài hiện tại  
void setLesson()  
{  
    message.setText("Bài học: " + mLesson);  
}  
  
}
```

Bạn có thể nhấn Shift + F10 để thay chữ trên máy ảo, ứng dụng hoạt động như kết quả trên hình.

Lưu lại và phục hồi trạng thái trong Activity

Ứng dụng trên hoạt động có vẻ không có vấn đề gì, tuy nhiên khi bấm đến bài số 5, thì màn hình có dòng chữ : **Bài học: 5** và nút bấm có nền GREEN. Đang ở trạng thái này, nếu bạn xoay ngang màn hình thì sau khi ứng dụng quay ngang (giả sử trường hợp này điện thoại bật chế độ cho phép xoay ngang) thì sau khi ứng dụng xoay ngang màn hình hiện thị kết quả không như cũ. Dòng chữ trở về nội dung **Bài học: 1** và màu nền nút bấm về màu mặc định.

Điều này xảy ra bởi vì trong vòng đời của Activity như chúng ta đã biết, khi Activity từ màn hình đứng quay ngang nó sẽ hủy Activity cũ (onPause -> onDestroy) rồi tạo Activity mới (gọi lại onCreate) do đó mà trạng thái sẽ trở về như lúc khởi tạo ban đầu.

Trường hợp nữa, khi Activity không còn hiện thị (onPause) nó nằm trong ngăn xếp hệ thống, nó có thể bị hệ thống Android hủy và khởi tạo lại bất kỳ lúc nào để quản lý tài nguyên (bộ nhớ), và thế thì khi Activity quay trở lại nó cũng không còn lưu trạng thái như trước khi đi vào onPause.

Vậy làm thế nào để khi Activity khởi tạo lại do hệ thống thay đổi (ví dụ quay ngang màn hình) mà trạng thái của nó vẫn giữ như trạng thái cũ?

Lưu lại trạng thái với onSaveInstanceState

Rất may là Activity có cơ chế để thực hiện việc này (lưu và phục hồi trạng thái). Trước tiên là khi Activity đi vào **onPause**, nó sẽ tự động gọi phương thức **onSaveInstanceState(Bundle outState)**, ta chỉ việc nạp chồng phương thức đó, và muốn lưu tạm trạng thái nào thì sẽ đưa vào **Bundle outState**

Ta sẽ nạp chồng(overridden) phương thức này để lưu lại giá trị của mLesson và màu của Button button

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    //Lưu lại giá trị mLesson với tên là "lesson" trong outState
    outState.putInt("lesson", mLesson);

    //Lấy màn nền button và Lưu Lại
    ColorDrawable colorButton = (ColorDrawable) button.getBackground();
    outState.putInt("buttoncolor", colorButton.getColor());
}
```

Bundle outState cho phép lưu dữ liệu trong nó theo cặp tên/giá trị bằng các phương thức: putInt(), putString(), putBoolean() ... tùy theo dữ liệu muốn lưu

Lưu ý khi nạp chồng **onSaveInstanceState** bạn cần gọi trong nó `super.onSaveInstanceState(outState);` để các View con trong giao diện cũng có thể lưu và phục hồi trạng thái riêng biệt (vì một số View như TextEdit cũng có cơ chế riêng để lưu và phục hồi trạng thái).

Phục hồi trạng thái ngay tại onCreate

Sau khi trạng thái (dữ liệu tạm) được lưu như trên, ta có thể đọc được khi Activity hoạt động ở vòng đời mới. Các giá trị ta lưu ở outState trong phương thức onSaveInstanceState sẽ quay trở lại onCreate trong savedInstanceState, vậy trong onCreate ta chỉ việc đọc trạng thái trước đây. Ví dụ phục hồi lại giá trị mLesson và màu button

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

    message = findViewById(R.id.message);
    button = findViewById(R.id.nextlesson);

    //Phục hồi
    if (savedInstanceState != null) {
        mLesson = savedInstanceState.getInt("lesson");
        button.setBackgroundColor(savedInstanceState.getInt("buttoncolor"));
    }
}
```

```
setLesson();

button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        mLesson++;
        setLesson();
        button.setBackgroundColor(Color.GREEN);
    }
});
}
```

Giờ ứng dụng của chúng ta đã lưu lại trạng thái, khi bạn thử xoay màn hình, sau khi Activity xoay nó sẽ phục hồi như màn hình trước khi xoay.

Trong đoạn mã trên, luôn cần kiểm tra (**savedInstanceState != null**) rồi mới đọc giá trị cần phục hồi bởi vì nếu Activity là khởi đầu mới (không phải kích hoạt lại) thì **savedInstanceState = null**. Nếu không muốn kiểm tra, có thể không phục hồi ở đây mà tại **onRestoreInstanceState**

Phục hồi trạng thái tại **onRestoreInstanceState(Bundle savedInstanceState)**

Mỗi khi **OnCreate** khởi chạy, nếu có trạng thái được lưu lại thì sau khi gọi **onCreate** Activity sẽ tự động gọi phương thức **onRestoreInstanceState(Bundle savedInstanceState)**, bạn cũng có thể quá tải phương thức này và đọc dữ liệu cần phục hồi tại **savedInstanceState**. Ví dụ:

```
@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);

    mLesson = savedInstanceState.getInt("lesson");
    button.setBackgroundColor(savedInstanceState.getInt("buttoncolor"));

    setLesson();
}
```

Kết quả cũng tương tự

AJ-04: Lưu và phục hồi trạ...



Sử dụng Bundle

Trong ví dụ trên, đã sử dụng đến đối tượng lớp **Bundle** để lưu và phục hồi trạng thái. Mục đích của **Bundle** để truyền dữ liệu giữa các Activity, nó được sử dụng cơ bản bởi **Intent**

Tạo một đối tượng **Bundle**

```
Bundle bundle = new Bundle();
```

Dữ liệu được đưa vào Bundle khá giống với Map, tức là dữ liệu nào đó sẽ được lưu theo cặp key và giá trị. Dữ liệu được đọc căn cứ vào key lưu trữ đó

Ví dụ:

```
bundle.putInt("songuyen", 1000); //key Là songuyen  
  
int giatri = bundle.getInt("songuyen"); //Đọc
```

Các phương thức để lưu dữ liệu như: **putInt, putBoolean, putString, putFloat ...** tương ứng có các phương thức đọc dữ liệu như **getInt, getBoolean, getString, getFloat ...**

Mục lục bài viết

[Tạo Project](#)[Lưu và phục hồi trạng thái](#)[Bundle](#)

Đăng ký theo dõi ứng hộ kênh



XuanThuLab

YouTube

Thích 1

Chia sẻ 1



0 bình luận

Sắp xếp theo **Mới nhất**

Thêm bình luận...

Plugin bình luận trên Facebook

[SQLite trong Android \(phần 2\) sử dụng SQLiteOpenHelper](#) [Sử dụng SQLite trong Android \(phần 1\)](#) [Sử dụng ListView hiển thị dữ liệu dạng danh sách trong Android](#) [Tạo và sử dụng Theme trong lập trình Android](#) [Thuộc tính style trong các View lập trình Android](#) [Các thành phần của một ứng dụng Android](#) [Cài đặt Android Studio, máy ảo Android, tạo dự án ứng dụng đầu tiên](#) [CoordinatorLayout \(2\) ứng dụng một số loại View](#) [Tìm hiểu các đơn vị px, dp, pt, in, mm, dip trong Androi](#) [CoordinatorLayout \(1\) xây dựng bố cục Behavior và Nested Scroll](#)

[Activity cơ bản](#)[View, ViewGroup, Layout](#)[Giới thiệu](#)[Privacy](#)[Lịch vạn niên](#) [Liên hệ](#) [RSS](#)

Đây là blog cá nhân, tôi ghi chép và chia sẻ những gì tôi học được ở đây về kiến thức lập trình PHP, Java, JavaScript, Android, C# ... và các kiến thức công nghệ khác

Developed by [XuanThuLab](#)

