***TỔ HỢP CẠC PHÍM TẮT TRONG ANDROID STUDIO***

Cửa sổ công cụ

Bạn có thể dùng phím tắt để mở cửa sổ công cụ. Bảng 1 liệt kê các phím tắt dùng cho các cửa sổ thường gặp nhất.

**Bảng 1.** Phím tắt cho các cửa sổ công cụ thường gặp

|  |  |  |
| --- | --- | --- |
| **Cửa sổ công cụ** | **Windows và Linux** | **macOS** |
| Dự án | Alt+1 | Command+1 |
| Quản lý phiên bản | Alt+9 | Command+9 |
| Chạy | Shift+F10 | Control+R |
| Gỡ lỗi | Shift+F9 | Control+D |
| Logcat | Alt+6 | Command+6 |
| Quay lại trình chỉnh sửa | Esc | Esc |
| Ẩn tất cả cửa sổ công cụ | Control+Shift+F12 | Command+Shift+F12 |

**Bảng 2.** Phím tắt để hoàn tất mã

|  |  |  |  |
| --- | --- | --- | --- |
| **Loại** | **Mô tả** | **Windows và Linux** | **macOS** |
| Hoàn tất cơ bản (Basic Completion) | Hiện các đề xuất cơ bản về biến, loại, phương thức, biểu thức, v.v. Nếu gọi kiểu hoàn thành cơ bản 2 lần liên tiếp, bạn sẽ thấy thêm nhiều kết quả, bao gồm cả các thành viên riêng tư và các thành viên tĩnh không nhập. | Control+Space | Control+Space |
| Hoàn tất thông minh (Smart Completion) | Hiện các tuỳ chọn liên quan dựa trên ngữ cảnh. Loại hình hoàn tất thông minh sẽ xem xét loại và các luồng dữ liệu dự kiến. Nếu gọi kiểu hoàn tất thông minh 2 lần liên tiếp, bạn sẽ thấy thêm nhiều kết quả, bao gồm cả các chuỗi. | Control+Shift+Space | Control+Shift+Space |
| Hoàn tất câu lệnh (Statement Completion) | Hoàn tất câu lệnh hiện tại, thêm dấu ngoặc đơn, dấu ngoặc vuông, dấu ngoặc nhọn còn thiếu, định dạng, v.v.. | Ctrl+Shift+Enter | Command+Shift+Enter |

* theo kiểu lạc đà), các đường dẫn, *line navigate*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Mô tả** | **Windows và Linux** | **macOS** |  |  |
| **Tác vụ chung** | | |  |  |
| Lưu tất cả | Ctrl+S | Command+S |  |  |
| Đồng bộ hoá | Ctrl+Alt+Y | Command+Option+Y |  |  |
| Phóng to/thu nhỏ trình chỉnh sửa | Ctrl+Shift+F12 | Ctrl+Command+F12 |  |  |
| Thêm vào mục ưa thích | Alt+Shift+F | Option+Shift+F |  |  |
| Kiểm tra tệp hiện có bằng hồ sơ hiện tại | Alt+Shift+I | Option+Shift+I |  |  |
| Lược đồ chuyển đổi nhanh | Ctrl+` (dấu phẩy ngược) | Ctrl+` (dấu phẩy ngược) |  |  |
| Mở hộp thoại cài đặt | Ctrl+Alt+S | Command+, (dấu phẩy) |  |  |
| Mở hộp thoại cấu trúc dự án | Ctrl+Alt+Shift+S | Command+; (dấu chấm phẩy) |  |  |
| Chuyển đổi giữa các thẻ và cửa sổ công cụ | Ctrl+Tab | Ctrl+Tab |  |  |
| **Di chuyển và tìm kiếm trong Android Studio** | | |  |  |
| Tìm kiếm mọi thứ (bao gồm cả mã và trình đơn) | Nhấn Shift hai lần | Nhấn Shift hai lần |  |  |
| Tìm | Ctrl+F | Command+F |  |  |
| Tìm mục tiếp theo | F3 | Command+G |  |  |
| Tìm mục trước đó | Shift+F3 | Command+Shift+G |  |  |
| Thay thế | Ctrl+R | Command+R |  |  |
| Tìm thao tác | Ctrl+Shift+A | Command+Shift+A |  |  |
| Tìm kiếm theo tên biểu tượng | Ctrl+Alt+Shift+N | Command+Option+O |  |  |
| Tìm lớp | Ctrl+N | Command+O |  |  |
| Tìm tệp (thay vì lớp) | Ctrl+Shift+N | Command+Shift+O |  |  |
| Tìm trong đường dẫn | Ctrl+Shift+F | Command+Shift+F |  |  |
| Mở hộp thoại cấu trúc tệp | Ctrl+F12 | Command+F12 |  |  |
| Chuyển giữa các thẻ trình chỉnh sửa đang mở | Alt+Mũi tên phải hoặc Mũi tên trái | Ctrl+Mũi tên phải hoặc Ctrl+Mũi tên trái |  |  |
| Chuyển ngay đến nguồn | F4 hoặc Ctrl+Enter | F4 hoặc Command+Mũi tên xuống |  |  |
| Mở thẻ trình chỉnh sửa hiện tại trong cửa sổ mới | Shift+F4 | Shift+F4 |  |  |
| Hộp thoại tệp mở gần đây | Ctrl+E | Command+E |  |  |
| Hộp thoại tệp được chỉnh sửa gần đây | Ctrl+Shift+E | Command+Shift+E |  |  |
| Chuyển đến vị trí chỉnh sửa gần đây nhất | Ctrl+Shift+Backspace | Command+Shift+Delete |  |  |
| Đóng thẻ trình chỉnh sửa đang hoạt động | Ctrl+F4 | Command+W |  |  |
| Quay lại cửa sổ trình chỉnh sửa trong cửa sổ công cụ | Esc | Esc |  |  |
| Ẩn cửa sổ công cụ đang hoạt động hoặc hoạt động gần đây nhất | Shift+Esc | Shift+Esc |  |  |
| Chuyển đến dòng | Ctrl+G | Command+L |  |  |
| Mở hệ phân cấp kiểu | Ctrl+H | Ctrl+H |  |  |
| Mở hệ phân cấp phương thức | Ctrl+Shift+H | Command+Shift+H |  |  |
| Mở hệ phân cấp lệnh gọi | Ctrl+Alt+H | Ctrl+Option+H |  |  |
| **Xem bố cục** | | |  |  |
| Phóng to/thu nhỏ | Ctrl+dấu cộng hoặc Ctrl+dấu trừ | Command+dấu cộng hoặc Command+dấu trừ |  |  |
| Vừa với màn hình | Ctrl+0 | Command+0 |  |  |
| Kích thước thực | Control+Shift+1 | Command+Shift+1 |  |  |
| **Công cụ thiết kế: Layout Editor (Trình chỉnh sửa bố cục)** | | |  |  |
| Chuyển đổi giữa chế độ Design (Thiết kế) và chế độ Blueprint (Bản thiết kế) | B | B |  |  |
| Chuyển đổi giữa chế độ Portrait (Dọc) và chế độ Landscape (Ngang) | O | O |  |  |
| Bật/tắt thiết bị | D | D |  |  |
| Buộc làm mới | R | R |  |  |
| Bật/tắt bảng lỗi kết xuất | E | E |  |  |
| Xoá điều kiện ràng buộc | Delete hoặc Control+nhấp chuột | Delete hoặc Command+nhấp |  |  |
| Phóng to | Ctrl+dấu cộng | Command+dấu cộng |  |  |
| Thu nhỏ | Ctrl+dấu trừ | Command+dấu trừ |  |  |
| Thu phóng cho vừa kích thước màn hình | Ctrl+0 | Command+0 |  |  |
| Kéo (hình ảnh) | Giữ Dấu cách+nhấp và kéo | Giữ Dấu cách+nhấp và kéo |  |  |
| Chuyển đến XML | Ctrl+B | Command+B |  |  |
| Chọn tất cả thành phần | Ctrl+A | Command+A |  |  |
| Chọn nhiều thành phần | Shift+nhấp hoặc Ctrl+nhấp | Shift+nhấp hoặc Command+nhấp |  |  |
| **Công cụ thiết kế: Navigation Editor (Trình chỉnh sửa điều hướng)** | | |  |  |
| Phóng to | Ctrl+dấu cộng | Command+dấu cộng |  |  |
| Thu nhỏ | Ctrl+dấu trừ | Command+dấu trừ |  |  |
| Thu phóng cho vừa kích thước màn hình | Ctrl+0 | Command+0 |  |  |
| Kéo (hình ảnh) | Giữ Dấu cách+nhấp và kéo | Giữ Dấu cách+nhấp và kéo |  |  |
| Chuyển đến XML | Ctrl+B | Command+B |  |  |
| Bật/tắt bảng lỗi kết xuất | E | E |  |  |
| Nhóm vào biểu đồ lồng nhau | Ctrl+G | Command+G |  |  |
| Đi qua các đích đến | Tab hoặc Shift+Tab | Tab hoặc Shift+Tab |  |  |
| Chọn tất cả đích đến | Ctrl+A | Command+A |  |  |
| Chọn nhiều đích đến | Shift+nhấp hoặc Ctrl+nhấp | Shift+nhấp hoặc Command+nhấp |  |  |
| **Viết mã** | | |  |  |
| Tạo mã (phương thức getter, setter, hàm khởi tạo, hashCode/equals, toString, tệp mới, lớp mới) | Alt+Insert | Command+N |  |  |
| Phương thức ghi đè | Ctrl+O | Ctrl+O |  |  |
| Phương thức triển khai | Ctrl+I | Ctrl+I |  |  |
| Bao quanh bằng (if...else, try...catch, v.v.) | Ctrl+Alt+T | Command+Option+T |  |  |
| Xoá dòng bằng con nháy | Control+Y | Command+Delete |  |  |
| Thu gọn/mở rộng khối mã hiện tại | Ctrl+dấu trừ hoặc Ctrl+dấu cộng | Command+dấu trừ hoặc Command+dấu cộng |  |  |
| Thu gọn/mở rộng tất cả khối mã | Ctrl+Shift+dấu trừ hoặc Ctrl+Shift+dấu cộng | Command+Shift+dấu trừ hoặc Command+Shift+plus |  |  |
| Sao chép dòng hoặc lựa chọn hiện tại | Ctrl+D | Command+D |  |  |
| Hoàn thành mã cơ bản | Control+Space | Control+Space |  |  |
| Hoàn tất mã thông minh (lọc danh sách biến và phương thức theo kiểu dự kiến) | Control+Shift+Space | Control+Shift+Space |  |  |
| Hoàn tất câu lệnh | Ctrl+Shift+Enter | Command+Shift+Enter |  |  |
| Tra cứu tài liệu nhanh | Ctrl+Q | Ctrl+J |  |  |
| Hiện tham số cho phương thức đã chọn | Ctrl+P | Command+P |  |  |
| Chuyển đến phần khai báo (trực tiếp) | Ctrl+B hoặc Ctrl+nhấp chuột | Command+B hoặc Command+nhấp chuột |  |  |
| Chuyển đến các tuỳ chọn triển khai | Ctrl+Alt+B | Command+Option+B |  |  |
| Chuyển đến phương thức mẹ (supermethod)/lớp mẹ (super-class) | Ctrl+U | Command+U |  |  |
| Mở phần tra cứu định nghĩa nhanh | Ctrl+Shift+I | Command+Y |  |  |
| Bật/tắt chế độ hiển thị cửa sổ công cụ dự án | Alt+1 | Command+1 |  |  |
| Chuyển đổi dấu trang | F11 | F3 |  |  |
| Chuyển đổi dấu trang bằng tính năng liên kết ghi nhớ | Control+F11 | Option+F3 |  |  |
| Nhận xét/huỷ nhận xét bằng nhận xét dòng | Control+/ | Command+/ |  |  |
| Nhận xét/huỷ nhận xét bằng nhận xét chặn | Control+Shift+/ | Command+Shift+/ |  |  |
| Chọn các khối mã tăng dần liên tiếp | Control+W | Option+Mũi tên lên |  |  |
| Giảm lựa chọn hiện tại về trạng thái trước đó | Ctrl+Shift+W | Option+Mũi tên xuống |  |  |
| Di chuyển đến đầu khối mã | Control+[ | Option+Command+[ |  |  |
| Di chuyển đến cuối khối mã | Control+] | Option+Command+] |  |  |
| Chọn đến đầu khối mã | Control+Shift+[ | Option+Command+Shift+[ |  |  |
| Chọn đến cuối khối mã | Control+Shift+] | Option+Command+Shift+] |  |  |
| Xoá đến cuối từ | Ctrl+Delete | Option+Delete |  |  |
| Xoá đến đầu từ | Ctrl+Backspace | Option+Delete |  |  |
| Tối ưu hoá nhập liệu | Ctrl+Alt+O | Ctrl+Option+O |  |  |
| Sửa lỗi nhanh trong dự án (cho thấy các thao tác theo ý định và thao tác sửa lỗi nhanh) | Alt+Enter | Option+Enter |  |  |
| Định dạng lại mã | Ctrl+Alt+L | Command+Option+L |  |  |
| Thụt lề tự động các dòng | Ctrl+Alt+I | Ctrl+Option+I |  |  |
| Thụt lề/bỏ thụt lề các dòng | Tab hoặc Shift+Tab | Tab hoặc Shift+Tab |  |  |
| Nối dòng thông minh | Ctrl+Shift+J | Ctrl+Shift+J |  |  |
| Tách dòng thông minh | Ctrl+Enter | Command+Enter |  |  |
| Bắt đầu dòng mới | Shift+Enter | Shift+Enter |  |  |
| Lỗi được đánh dấu tiếp theo/trước | F2 hoặc Shift+F2 | F2 hoặc Shift+F2 |  |  |
| **Xây dựng và chạy** | | |  |  |
| Xây dựng | Ctrl+F9 | Command+F9 |  |  |
| Xây dựng và chạy | Shift+F10 | Ctrl+R |  |  |
| Áp dụng các thay đổi và khởi động lại hoạt động | Control+F10 | Ctrl+Command+R |  |  |
| Áp dụng thay đổi mã | Ctrl+Alt+F10 | Control+Command+Shift+R |  |  |
| **Gỡ lỗi** | | |  |  |
| Gỡ lỗi | Shift+F9 | Ctrl+D |  |  |
| Nhảy cóc | F8 | F8 |  |  |
| Nhảy vào | F7 | F7 |  |  |
| Nhảy vào thông minh | Shift+F7 | Shift+F7 |  |  |
| Nhảy ra | Shift+F8 | Shift+F8 |  |  |
| Đến con trỏ | Alt+F9 | Option+F9 |  |  |
| Đánh giá biểu thức | Alt+F8 | Option+F8 |  |  |
| Tiếp tục chương trình | F9 | Command+Option+R |  |  |
| Bật/tắt điểm ngắt | Control+F8 | Command+F8 |  |  |
| Xem điểm ngắt | Control+Shift+F8 | Command+Shift+F8 |  |  |
| **Tái cấu trúc** | | |  |  |
| Sao chép | H5 | H5 |  |  |
| Di chuyển | F6 | F6 |  |  |
| Xoá an toàn | Alt+Delete | Command+Delete |  |  |
| Đổi tên | Shift+F6 | Shift+F6 |  |  |
| Thay đổi chữ ký | Control+F6 | Command+F6 |  |  |
| Cùng dòng | Control+Alt+N | Command+Option+N |  |  |
| Phương pháp trích xuất | Ctrl+Alt+M | Command+Option+M |  |  |
| Trích xuất biến | Ctrl+Alt+V | Command+Option+V |  |  |
| Trích xuất trường | Ctrl+Alt+F | Command+Option+F |  |  |
| Trích xuất hằng số | Ctrl+Alt+C | Command+Option+C |  |  |
| Trích xuất tham số | Ctrl+Alt+P | Command+Option+P |  |  |
| **Quản lý phiên bản/nhật ký trên máy** | | |  |  |
| Cam kết dự án với VCS | Ctrl+K | Command+K |  |  |
| Cập nhật dự án từ VCS | Ctrl+T | Command+T |  |  |
| Xem các thay đổi gần đây | Alt+Shift+C | Option+Shift+C |  |  |
| Mở hộp thoại VCS | Alt+` (dấu phẩy ngược) | Ctrl+V |  |  |

**I. KHÁI QUÁT VỀ CONTEXT**

**Context là thành phần trong ứng dụng android cung cấp quyền truy cập thông tin về các trạng thái của ứng dụng đó. Nó cung cấp các Activities, Fragments và Services truy cập tới các file tài nguyên, hình ảnh, theme, style và các file nằm ngoài ứng dụng. Nó cũng cho phép truy cập vào các thành phần chính của Android như layout, keyboard và tìm kiếm các content providers.**

### **ĐỊNH NGHĨA:**

### **LayoutInflater là 1 component giúp bạn chuyển layout file(Xml) thành View(Java code) trong Android. Bạn thường sử dụng nó trong phương thức onCreateView của fragment hoặc phương thức getView khi custom adapter.**

### **PHƯƠNG THỨC INFLATE**

Công việc của LayoutInflater là đọc xml layout file và chuyển đổi các thuộc tính của nó thành 1 View trong Java code. Sau khi có đối tượng LayoutInflater, ta có thể dùng phương thức inflate để chuyển đổi 1 xml layout file thành 1 View trong java. Ta có 2 phương thức inflate với số lượng tham số khác nhau:

1. View view = layoutInflater.inflate(int resource, ViewGroup parent)2. View view = layoutInflater.inflate(int resource, ViewGroup parent, boolean attachToRoot)

Các bạn sẽ thắc mắc các tham số của inflater có ý nghĩa gì? 2 Phương thức inflate trên chỉ khác nhau tham số attachToRoot vậy attachToRoot là gì? Cùng tìm hiểu thông qua 1 số ví dụ nhé.  
  
Trước tiên chúng ta tìm hiểu 3 tham số của nó là gì đã nhé: Như định nghĩa thì nhiệm vụ của LayoutInflater là chuyển đổi xml layout file thành đối tượng View trong java code, vậy thì:

* Tham số thứ nhất là: int resource, nó chính là xml layout file mà chúng ta muốn chuyển đổi thành View.
* Tham số thứ hai là: ViewGroup parent, nó là ViewGroup nơi mà xml layout file(tham số thứ nhất) có thể được nhúng vào, LayoutInflater sẽ chuyển đổi xml layout file thành View và sử dụng các thuộc tính phù hợp với ViewGroup parrent.
* Tham số thứ ba là: attachToRoot, khi mà attachToRoot=true thì ngay sau khi quá trình chuyển đổi xml file(resource) thành View hoàn thành thì nó sẽ nhúng View đó vào ViewGroup parent (****RIGHT NOW****) , khi attachToRoot = false thì nó chỉ chuyển đổi xml file(resource) thành View trong java mà không thêm ngay vào ViewGroup(****NOT NOW****)

**SHOWASACTION**

Nếu menu của bạn cung cấp cả tiêu đề (title) và icon sử dụng thuộc tính title và icon trong XML, sau đấy menu sẽ chỉ show icon theo mặc định, Nếu bạn muốn hiển thị thêm tiêu đề, thêm thuộc tính “widthText” tới showAsAction. Ví dụ:

<item yourapp:showAsAction="ifRoom|withText" ... />

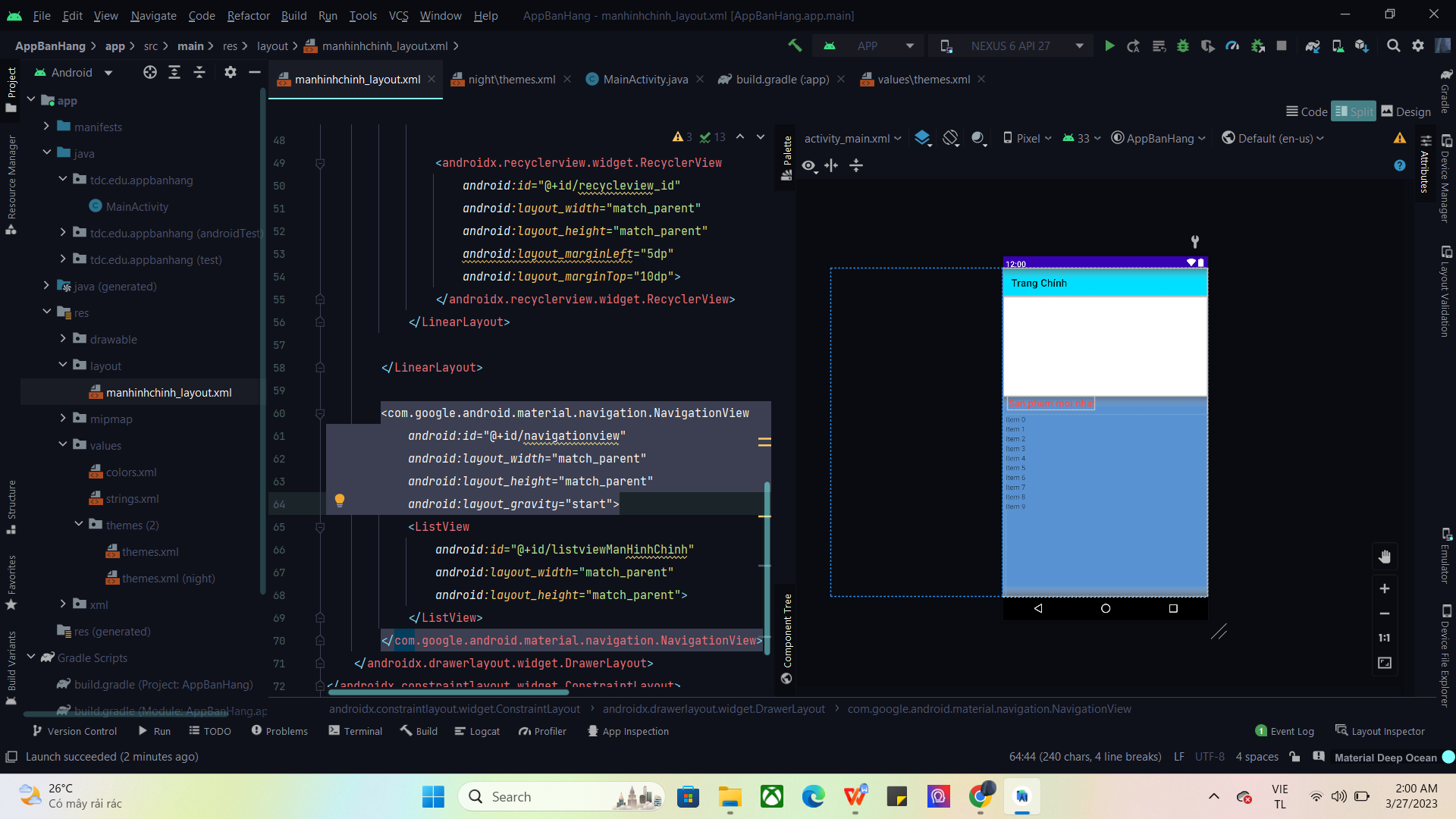
****Chú ý****: Các “withText” giá trị là một gợi ý (hint) cho các Action Bar rằng các tiêu đề văn bản sẽ xuất hiện. Action Bar sẽ hiển thị tiêu đề khi có thể, nhưng có thể không nếu một icon có sẵn và Action Bar bị hạn chế về không gian.

Để truy xuất chiều cao của ActionBar trong XML, chỉ cần sử dụng

?android:attr/actionBarSize

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<com.google.android.material.navigation.NavigationView  
 *android:id*="@+id/navigationview"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent"  
 *android:layout\_gravity*="start"/> //layout\_gravity : nơi nó nằmg vd: start nằm bên tay trái màn hình chính ẩn ra ngoài.



Draw chứa navigation

Bắt sự kiện và chuyển navigation ra giữa màn hình dựa trên Drawlayout.

*private void* Actionbar() {  
 setSupportActionBar(toolbar\_ManHinhChinh);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(*true*);  
 toolbar\_ManHinhChinh.setNavigationIcon(R.drawable.***left\_menu***);  
*// Catch clip action* toolbar\_ManHinhChinh.setNavigationOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View view) {  
*// Move to center screnn* drawerLayout.openDrawer(GravityCompat.***START***);  
 }  
 });

Cho phép hiện ra cái icon

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**RECYCLEVIEW**

**MYRECYCLEVIEW**

*public class* MyRecycleView *extends* RecyclerView.Adapter<MyRecycleView.MyViewHolder> {  
 *private* Activity activity;  
 *private int* Layout\_Id;  
 *private* ArrayList<tdc.edu.myrecycleview.model.Person> personArrayList;  
 *//B2:Create variable of this Interface.  
 private OnRecycleViewItemClickListener* onRecycleViewItemClickListenerVar;  
  
 *//Constructor of this class  
 public* MyRecycleView(Activity activity, *int* layout\_Id, ArrayList<tdc.edu.myrecycleview.model.Person> personArrayList) {  
 *this*.activity = activity;  
 Layout\_Id = layout\_Id;  
 *this*.personArrayList = personArrayList;  
 }  
  
 *//When you extends MyViewHolder you will have 3 method need your definitions.* @NonNull  
 @Override  
 *public* MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, *int* viewType) {  
 LayoutInflater inflater = activity.getLayoutInflater();  
 CardView view = (CardView) inflater.inflate(viewType, parent, *false*);  
 *return new* MyViewHolder(view);  
 }  
 *//Do du lieu vao holder* @Override  
 *public void* onBindViewHolder(@NonNull MyViewHolder holder, *int* position) {  
 tdc.edu.myrecycleview.model.Person person = personArrayList.get(position);  
 holder.txt\_name.setText(person.getName());  
 holder.txt\_hobbies.setText(person.getHobbies());

viewHolder.onClickListener = *new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View v) {  
 *//Log.d("onBinding", "Called " + position);  
 if*(recylerItemViewClickListenner != *null*){  
 recylerItemViewClickListenner.onItemClickListenner(position,viewHolder.itemView);  
 }  
 }  
};

}  
  
 @Override  
 *public int* getItemCount() {  
 *return* personArrayList.size();  
 }  
  
 Override lại phương thức getItemViewType(position: Int)

để get type tương ứng với từng position.  
 @Override  
 *public int* getItemViewType(*int* position) {  
 *return* Layout\_Id;  
 }  
  
 *//B1: Definition MyViewHolder class  
 protected static class* MyViewHolder *extends* RecyclerView.ViewHolder *implements* View.*OnClickListener* {  
 ImageView imageView;  
 TextView txt\_name;  
 TextView txt\_hobbies;  
 View.*OnClickListener* onClickListener;  
  
 *public* MyViewHolder(@NonNull View itemView) {  
 *super*(itemView);  
 imageView = itemView.findViewById(R.id.***ImageView\_ID***);  
 txt\_name = itemView.findViewById(R.id.***TextViewName\_ID***);  
 txt\_hobbies = itemView.findViewById(R.id.***TextViewHobbie\_ID***);  
  
 imageView.setOnClickListener(*this*);  
 txt\_name.setOnClickListener(*this*);  
 txt\_hobbies.setOnClickListener(*this*);  
 itemView.setOnClickListener(*this*);  
 }  
  
 @Override  
 *public void* onClick(View view) {  
 *if* (onClickListener != *null*) {  
 onClickListener.onClick(view);  
 }  
 }  
  
 }  
  
 *//B1: Definitions interface  
 public interface OnRecycleViewItemClickListener* {  
 *public void* onItemClickListener(*int* positions, View cardView);  
 }  
  
 *//:B4 setter for variable  
 public void* setOnRecycleViewItemClickListenerVar(*OnRecycleViewItemClickListener* onRecycleViewItemClickListenerVar) {  
 *this*.onRecycleViewItemClickListenerVar = onRecycleViewItemClickListenerVar;  
 }  
}

MAIN\_ACTIVITY

@Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 listView = findViewById(R.id.***RecyclerView\_ID***);  
 ArrayList<Person> personArrayList = *new* ArrayList<>();  
 Person person1 = *new* Person("Chu dinh hanh", "choi xe");  
 Person person2 = *new* Person("Chu dinh hanh", "choi xe");  
 Person person3 = *new* Person("Chu dinh hanh", "choi xe");  
 Person person4 = *new* Person("Chu dinh hanh", "choi xe");  
 Person person5 = *new* Person("Chu dinh hanh", "choi xe");  
 personArrayList.add(person1);  
 personArrayList.add(person2);  
 personArrayList.add(person3);  
 personArrayList.add(person4);  
 personArrayList.add(person5);  
 MyRecycleView myRecycleView = *new* MyRecycleView(*this*, R.layout.***customize\_recyclerview\_item***, personArrayList);  
*// //Create LinearLayout manager* LinearLayoutManager layoutManager = *new* LinearLayoutManager(*this*);  
 layoutManager.setOrientation(LinearLayoutManager.***HORIZONTAL***);  
 listView.setLayoutManager(layoutManager);  
 listView.setAdapter(myRecycleView);  
 }  
}

customize\_recyclerview\_item./ phải chứa trong cardview.

<androidx.cardview.widget.CardView>

<LinearLayout>

Activity\_main.xml

<LinearLayout>  
 <androidx.recyclerview.widget.RecyclerView />  
</LinearLayout>

1 SỐ ĐIỀU CẦN CHÚ Ý.

<?*xml version*="1.0" *encoding*="utf-8"?>  
<androidx.cardview.widget.CardView  
 *xmlns:android*="http://schemas.android.com/apk/res/android"  
 *xmlns:card\_view*="http://schemas.android.com/apk/res-auto"  
  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="wrap\_content"  
 *android:orientation*="horizontal"

*Tạo khoảng cách cho item vơi giao diện màn hình*

*card\_view:cardUseCompatPadding*="true"

được sử dụng để xác định kích thước và độ mềm của bóng để mô tả độ sâu một cách chân thực.  
 *card\_view:cardElevation*="5dp"  
 *card\_view:cardCornerRadius*="5dp">

MainActivity

myRecycleView.set\_onRecycleViewItemClickListener(*new* MyRecycleView.OnRecycleViewItemClickListener() {  
 @Override  
 *public void* onItemClickListener() {  
   
 }  
});

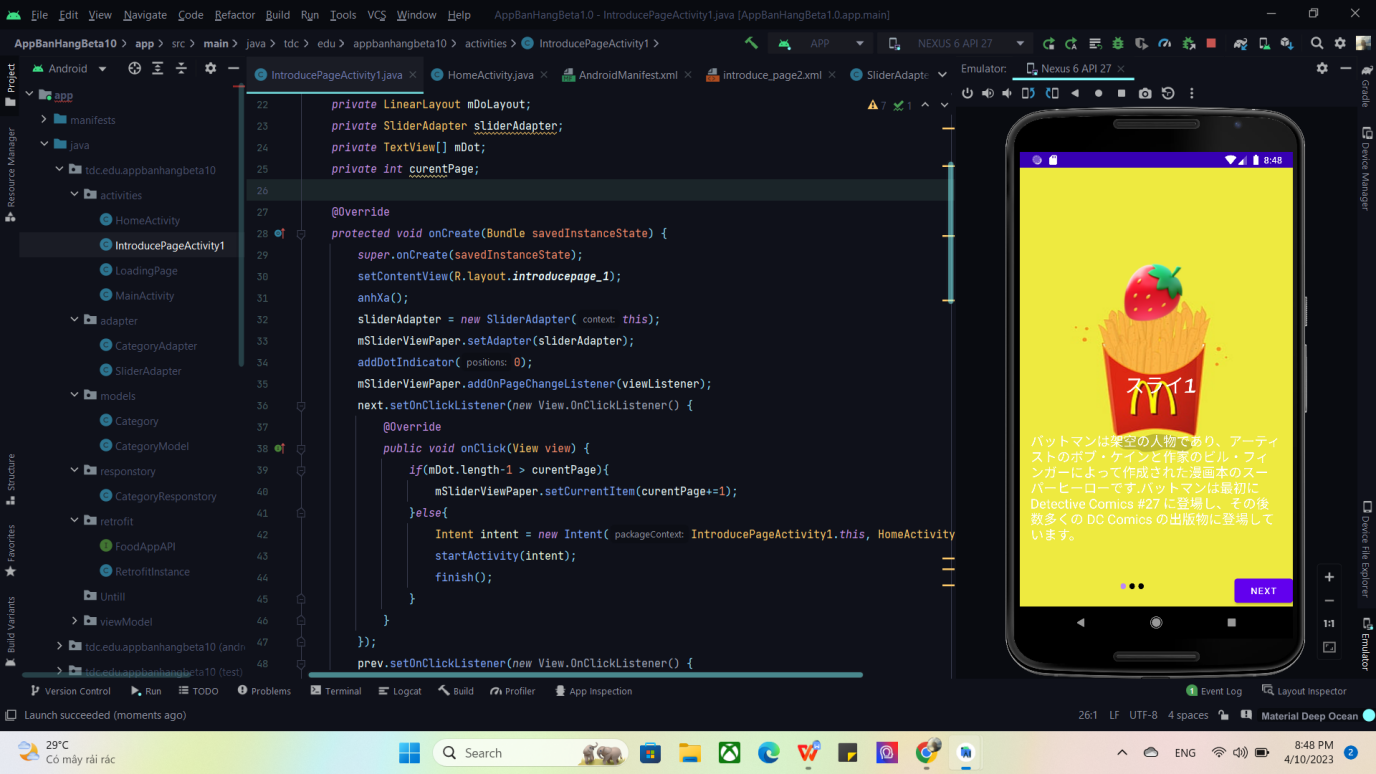
**SPINNER**

*public void* setContextForSpinner() {  
 ArrayList<String> arrayList = *new* ArrayList<>();  
 arrayList.add("HA NOI");  
 arrayList.add("SAI GON");  
 arrayList.add("HAI PHONG");  
 arrayList.add("PHNOMPENK");  
 arrayList.add("JAKATTA");  
 ArrayAdapter spinnerArray = *new* ArrayAdapter(*this*, androidx.appcompat.R.layout.*support\_simple\_spinner\_dropdown\_item*, arrayList);  
 spinner.setAdapter(spinnerArray);  
}

**NAVIGATION BOTTOM**

<com.google.android.material.bottomappbar.BottomAppBar  
 *android:id*="@+id/appbar"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_gravity*="bottom"  
 *app:fabCradleMargin*="@dimen/\_4sdp"  
 *app:fabCradleVerticalOffset*="0dp"  
 *app:fabCradleRoundedCornerRadius*="@dimen/\_10sdp">

**VIEWPAYER LAYOUT**

<?*xml version*="1.0" *encoding*="utf-8"?>  
<RelativeLayout *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"  
 *android:background*="@drawable/androidhome"  
 *tools:context*=".activities.IntroducePageActivity1">  
  
  
 <androidx.viewpager.widget.ViewPager  
 *android:id*="@+id/mSliderViewPaper"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="601dp"  
 *android:layout\_alignParentTop*="true"  
 *android:orientation*="horizontal">  
 </androidx.viewpager.widget.ViewPager>  
  
 <LinearLayout  
 *android:id*="@+id/mDolayout"  
 *android:layout\_width*="118dp"  
 *android:layout\_height*="54dp"  
 *android:layout\_below*="@+id/mSliderViewPaper"  
 *android:layout\_alignParentStart*="true"  
 *android:layout\_alignParentEnd*="true"  
 *android:layout\_alignParentBottom*="true"  
 *android:layout\_marginStart*="149dp"  
 *android:layout\_marginEnd*="144dp"  
 *android:layout\_marginBottom*="4dp"  
 *android:orientation*="horizontal"></LinearLayout>  
  
 <Button  
 *android:visibility*="invisible"  
 *android:id*="@+id/btn\_prev"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_alignParentStart*="true"  
 *android:layout\_alignParentBottom*="true"  
 *android:text*="Previous" />  
 <Button  
 *android:id*="@+id/btn\_next"  
 *android:text*="Next"  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:layout\_alignParentEnd*="true"  
 *android:layout\_alignParentBottom*="true" />  
</RelativeLayout>

**ITEMVIEWPAYER LAYOUT**

<?*xml version*="1.0" *encoding*="utf-8"?>  
<RelativeLayout *xmlns:android*="http://schemas.android.com/apk/res/android"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent">  
  
 <ImageView  
 *android:id*="@+id/ImageView\_Main\_ID"  
 *android:layout\_width*="207dp"  
 *android:layout\_height*="167dp"  
 *android:layout\_alignParentStart*="true"  
 *android:layout\_alignParentTop*="true"  
 *android:layout\_alignParentEnd*="true"  
 *android:layout\_alignParentBottom*="true"  
 *android:layout\_marginStart*="96dp"  
 *android:layout\_marginTop*="131dp"  
 *android:layout\_marginEnd*="108dp"  
 *android:layout\_marginBottom*="361dp"  
 *android:src*="@drawable/cherry" />  
  
 <TextView  
 *android:fontFamily*="@font/font1"  
 *android:id*="@+id/txt\_heading"  
 *android:layout\_width*="133dp"  
 *android:layout\_height*="52dp"  
 *android:layout\_below*="@+id/ImageView\_Main\_ID"  
 *android:layout\_alignParentStart*="true"  
 *android:layout\_alignParentEnd*="true"  
 *android:layout\_marginStart*="145dp"  
 *android:layout\_marginTop*="-297dp"  
 *android:layout\_marginEnd*="133dp"  
 *android:gravity*="center"  
 *android:text*="HEADING 1"  
 *android:textColor*="@color/white"  
 *android:textSize*="30dp" />  
  
 <TextView  
 *android:id*="@+id/txt\_content"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="220dp"  
 *android:layout\_below*="@+id/txt\_heading"  
 *android:layout\_alignParentBottom*="true"  
 *android:layout\_marginTop*="25dp"  
 *android:layout\_marginBottom*="0dp"  
 *android:padding*="16dp"  
 *android:text*="Pinocchio là một bộ phim truyền hình Hàn Quốc năm 2014 với diễn viên chính Lee Jong-sukvà Park Shin-hye. Bộ phim được phát sóng trên SBS vào thứ tư và thứ năm lúc 21:55 dài 20 tập và bắt đầu từ 12 tháng 11 năm 2014."  
 *android:textColor*="@color/white"  
 *android:textSize*="20sp" />  
  
  
</RelativeLayout>

**ADAPTER**

*package* tdc.edu.appbanhangbeta10.adapter;  
  
*import* android.app.Activity;  
*import* android.content.Context;  
*import* android.view.LayoutInflater;  
*import* android.view.View;  
*import* android.view.ViewGroup;  
*import* android.widget.ImageView;  
*import* android.widget.LinearLayout;  
*import* android.widget.RelativeLayout;  
*import* android.widget.TextView;  
  
*import* androidx.annotation.NonNull;  
*import* androidx.viewpager.widget.PagerAdapter;  
  
*import* tdc.edu.appbanhangbeta10.R;  
  
*public class* SliderAdapter *extends* PagerAdapter {  
  
 Context context;  
 LayoutInflater layoutInflater;  
  
 *public* SliderAdapter(Context context) {  
 *this*.context = context;  
 }  
  
 *public int*[] slider\_images = {  
 R.drawable.***strawberry***,  
 R.drawable.***blueberry***,  
 R.drawable.***grapes*** };  
  
  
 *public* String[] slide\_Heading = {  
 "スライ1",  
 "スライ2",  
 "スライ3"  
 };  
  
  
 *public* String[] slide\_string = {  
 "バットマンは架空の人物であり、アーティストのボブ・ケインと作家のビル・フィンガーによって作成された漫画本のスーパーヒーローです.バットマンは最初に Detective Comics #27 に登場し、その後数多くの DC Comics の出版物に登場しています。",  
 "スパイダーマンは、マーベル コミックが発行するコミックに登場する架空のスーパーヒーローです。このキャラクターは、ライターのスタン・リーとライター兼アーティストのスティーブ・ディッコによって作成され、アメイジング・ファンタジー #15 に初登場しました。",  
 "スーパーマンは、DC コミックスが発行する同名の有名なアメリカン コミック シリーズに登場する架空のスーパー ヒーロー キャラクターです。スーパーマンは、1933 年にアメリカの作家ジェリー シーゲルとカナダ生まれのアーティスト ジョー シャスターによって当時オハイオ州クリーブランドに住んでいた高校生によって作成されました。"  
 };  
  
 @Override  
 *public int* getCount() {  
 *return* slide\_Heading.length;  
 }  
  
 @Override  
 *public boolean* isViewFromObject(@NonNull View view, @NonNull Object object) {  
 *return* view == (RelativeLayout) object;  
 }  
  
 @NonNull  
 @Override  
 *public* Object instantiateItem(@NonNull ViewGroup container, *int* position) {  
 layoutInflater = (LayoutInflater) context.getSystemService(context.***LAYOUT\_INFLATER\_SERVICE***);  
 View view = layoutInflater.inflate(R.layout.***introduce\_page2***, container, *false*);  
 ImageView imageView = view.findViewById(R.id.***ImageView\_Main\_ID***);  
 TextView txt\_heading = view.findViewById(R.id.***txt\_heading***);  
 TextView txt\_content = view.findViewById(R.id.***txt\_content***);  
  
 imageView.setImageResource(slider\_images[position]);  
 txt\_heading.setText(slide\_Heading[position]);  
 txt\_content.setText(slide\_string[position]);  
  
 container.addView(view);  
 *return* view;  
 }  
  
 @Override  
 *public void* destroyItem(@NonNull ViewGroup container, *int* position, @NonNull Object object) {  
 container.removeView((RelativeLayout) object);  
 }  
}

**ACTIVITY**

*package* tdc.edu.appbanhangbeta10.activities;  
  
*import* androidx.appcompat.app.AppCompatActivity;  
*import* androidx.viewpager.widget.ViewPager;  
  
*import* android.content.Intent;  
*import* android.os.Bundle;  
*import* android.text.Html;  
*import* android.view.View;  
*import* android.widget.Button;  
*import* android.widget.LinearLayout;  
*import* android.widget.TextView;  
*import* android.widget.Toast;  
  
*import* tdc.edu.appbanhangbeta10.R;  
*import* tdc.edu.appbanhangbeta10.adapter.SliderAdapter;  
  
*public class* IntroducePageActivity1 *extends* AppCompatActivity {  
  
 *private* Button prev, next;  
 *private* ViewPager mSliderViewPaper;  
 *private* LinearLayout mDoLayout;  
 *private* SliderAdapter sliderAdapter;  
 *private* TextView[] mDot;  
 *private int* curentPage;  
  
 @Override  
 *protected void* onCreate(Bundle savedInstanceState) {  
 *super*.onCreate(savedInstanceState);  
 setContentView(R.layout.***introducepage\_1***);  
 anhXa();  
 sliderAdapter = *new* SliderAdapter(*this*);  
 mSliderViewPaper.setAdapter(sliderAdapter);  
 addDotIndicator(0);  
 mSliderViewPaper.addOnPageChangeListener(viewListener);  
 next.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View view) {  
 *if*(mDot.length-1 > curentPage){  
 mSliderViewPaper.setCurrentItem(curentPage+=1);  
 }*else*{  
 Intent intent = *new* Intent(IntroducePageActivity1.*this*, HomeActivity.*class*);  
 startActivity(intent);  
 finish();  
 }  
 }  
 });  
 prev.setOnClickListener(*new* View.OnClickListener() {  
 @Override  
 *public void* onClick(View view) {  
 mSliderViewPaper.setCurrentItem(curentPage-=1);  
 }  
 });  
 }  
  
 *public void* anhXa() {  
 prev = findViewById(R.id.***btn\_prev***);  
 next = findViewById(R.id.***btn\_next***);  
 mSliderViewPaper = findViewById(R.id.***mSliderViewPaper***);  
 mDoLayout = findViewById(R.id.***mDolayout***);  
 }  
  
 *public void* addDotIndicator(*int* positions) {  
 mDot = *new* TextView[3];  
 mDoLayout.removeAllViews();  
 *for* (*int* i = 0; i < mDot.length; i++) {  
 mDot[i] = *new* TextView(*this*);  
 mDot[i].setText(Html.*fromHtml*("&#8226;"));  
 mDot[i].setTextSize(40);  
 mDot[i].setTextColor(getResources().getColor(R.color.***black***));  
 mDoLayout.addView(mDot[i]);  
 }  
 *if* (mDot.length > 0) {  
 mDot[positions].setTextColor(getResources().getColor(R.color.***purple\_200***));  
 }  
 }  
  
 ViewPager.*OnPageChangeListener* viewListener = *new* ViewPager.OnPageChangeListener() {  
 @Override  
 *public void* onPageScrolled(*int* position, *float* positionOffset, *int* positionOffsetPixels) {  
 }  
  
 @Override  
 *public void* onPageSelected(*int* position) {  
 addDotIndicator(position);  
 *if* (position == 0) {  
 next.setEnabled(*true*);  
 prev.setEnabled(*false*);  
 prev.setText("");  
 prev.setVisibility(View.***INVISIBLE***);  
 next.setVisibility(View.***VISIBLE***);  
  
 } *else if* (mDot.length-1 == position) {  
 next.setEnabled(*true*);  
 next.setVisibility(View.***VISIBLE***);  
 prev.setEnabled(*true*);  
 prev.setVisibility(View.***VISIBLE***);  
 next.setText("FINISH");  
 } *else* {  
 next.setText("NEXT");  
 prev.setEnabled(*true*);  
 next.setEnabled(*true*);  
 prev.setText("PREVIOUS");  
 prev.setVisibility(View.***VISIBLE***);  
 next.setVisibility(View.***VISIBLE***);  
 }  
 }  
  
 @Override  
 *public void* onPageScrollStateChanged(*int* state) {  
  
 }  
 };  
}

**Navigations**

<?*xml version*="1.0" *encoding*="utf-8"?>  
<LinearLayout *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*=".activity.ManHinhChinh">  
  
 <androidx.drawerlayout.widget.DrawerLayout  
 *android:id*="@+id/DrawerLayout\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent">  
  
 <LinearLayout  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent"  
 *android:orientation*="vertical">  
  
 <androidx.appcompat.widget.Toolbar  
 *android:id*="@+id/Toolbar\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="?attr/actionBarSize"  
 *android:background*="#F44336"  
 *app:title*="Trang Chinh">  
  
 </androidx.appcompat.widget.Toolbar>  
 *<!-- Slide quang cao -->* <ViewFlipper  
 *android:id*="@+id/ViewFlipper\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="200dp">  
 </ViewFlipper>  
 *<!-- danh sach san pham-->* <LinearLayout  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent"  
 *android:background*="#F44336"  
 *android:orientation*="vertical">  
  
 <TextView  
 *android:layout\_width*="wrap\_content"  
 *android:layout\_height*="wrap\_content"  
 *android:text*="San pham moi nhat"  
 *android:textColor*="@color/white"  
 *android:textSize*="20sp"  
 *android:textStyle*="bold" />  
  
 <androidx.recyclerview.widget.RecyclerView  
 *android:id*="@+id/RecyclerView\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent">  
  
 </androidx.recyclerview.widget.RecyclerView>  
 </LinearLayout>  
 </LinearLayout>  
 <com.google.android.material.navigation.NavigationView  
 *android:id*="@+id/NavigationView\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent"  
 *android:layout\_gravity*="start">  
  
 <ListView  
 *android:id*="@+id/ListView\_manhinhchinh"  
 *android:layout\_width*="match\_parent"  
 *android:layout\_height*="match\_parent" />  
  
 </com.google.android.material.navigation.NavigationView>  
  
 </androidx.drawerlayout.widget.DrawerLayout>  
  
  
</LinearLayout>