2019ver

textview

<TextView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="textview 예제"  
 />

Match\_parent 는 부모 크기만큼

Wrap\_content 는 내용물의 크기만큼

Linear레이아웃 에는 orientation을 붙여주어야 한다.

Vertical은 세로 horizen은 가로이다.

Textcolor =”#ffffff” 를통해 색상을 지정할 수 있고 왼쪽의 박스를 눌러 원하는 색상을 지;정할 수 있다.

Textsize=”nsp”로 크지정이 가능하다.

Edittext 의 경우

Hint를 넣으면 입력 전에 가표시를 할 수 있다.

Button

Text로 버튼의 텍스트를 정할 수 있다.

Id 엔터 엔터 한 뒤 /뒤에 원하는 아이디 값을지정할 수 있다.

main에서 사용하기

et\_id = findViewById(R.id.*et\_id*);  
//아이디를 통해 불러옴  
btn\_test = findViewById(R.id.*btn\_test*);  
  
//버튼 클릭시의 처리  
btn\_test.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 //버튼의 텍스트를 edittext 의 글씨를 가져와서 버튼의 글자로 지정  
 btn\_test.setText(et\_id.getText());  
  
 //edittext창의 글자를 바꿈  
 et\_id.setText("입력됨");  
 }  
});

intent – 화면전환

activity = 화면전환

a라는 activity에서 b라는 activity로 이동.

mainactivity폴더에 subactivity를 만든다

버튼을 누르면 이동 하는 예제

btn\_move.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Intent intent = new Intent(MainActivity.this,SubActivity.class);  
 startActivity(intent);//액티비티 이동  
  
 }  
});

Intent 를 생성할 때 첫 인자로 현재 activity 두번째 인자로 이동할 activity를 선택한다.

다음 화면으로 데이터를 넘길 때

intent.putExtra("str",str);

key 와 value형태로 putExtra를 통해 넣는다

받는쪽

Intent intent = getIntent();  
String str = intent.getStringExtra("str");  
tv\_text = findViewById(R.id.*tv\_text*);  
tv\_text.setText(str);

getIntent로 intent를 통한 화면전환시 intent를 받을 수 있고,

getStringExtra등 해당 자료형을 key를 지정하여 받을 수 있다.

ImageView 와 Toast 메시지

<ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:src="@mipmap/ic\_launcher"  
 android:id="@+id/test"  
 />

이미지 뷰를 생성하고 src를 통해 이미지의 주소를 작성한다. Mipmap은 아이콘 등의 이미지가 저장되는 기본 위치이다.

이미지 클릭시 Toast 메시지 출력

test = findViewById(R.id.*test*);  
test.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 Toast.*makeText*(getApplicationContext(), "배고파", Toast.*LENGTH\_SHORT*).show();  
 }  
});

첫 인자는 현재 activity화면을 의미하고 두번째는 표시할 텍스트, 세번째는 표시될 길이이다.

.show()를 통해 메시지를 출력한다.

정렬하기

gravity속성을 통해 정렬한다.

android:gravity는 위젯자체로의 정렬 값을 지정

android:layout\_gravity는 위젯자체가 아닌 위젯을 감싸주고있는 레이아웃의 정렬 값을 지정

리니어 레이아웃으로 감싼 뒤 해당 레이아웃을 정렬하는 방식사용.

<LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:gravity="center"  
 >  
  
 <ImageView  
 android:layout\_width="100dp"  
 android:layout\_height="100dp"  
 android:src="@mipmap/ic\_launcher"  
 android:id="@+id/test"  
 />  
</LinearLayout>

ListView

리스트 뷰 선언 자체는 간단하지만 사용할 activity에서 만져줘야함

//리스트 뷰를 가져온다  
list = findViewById(R.id.*list*);  
  
//리스트 뷰에 넣을 목록을 생성  
List<String> data = new ArrayList<>();  
  
//리스트 뷰와 리스트 를 연결해주는 어댑터  
//첫 인자로 해당 acitivity 두번째 인자로 ui, 세번째 인자로 들어갈 항목이다  
ArrayAdapter<String> adapter = new ArrayAdapter<>(this, android.R.layout.*simple\_list\_item\_1*,data);  
  
//리스트 뷰에 adapter를 설정한다.  
list.setAdapter(adapter);  
  
data.add("갸아아앙");  
data.add("햐아아악");  
data.add("먀아아아악");  
  
//adapter에 설정을 저장한다.  
adapter.notifyDataSetChanged();

String[] items = {"망고쥬스","토마토쥬스","포도쥬스"};  
ListAdapter adapter = new ArrayAdapter<String>(this, android.R.layout.*simple\_list\_item\_1*,items);  
ListView listView = (ListView)findViewById(R.id.*listview*);  
listView.setAdapter(adapter);

이와 같은 방식도 가능하다.

Navigation menu bar

프로젝트 생성시

Navigation drawer activity를 선택한다

버전의 차이로 예제처럼 하려면

implements NavigationView.OnNavigationItemSelectedListener

를 추가한뒤 메소드를 상속받고

@Override  
public boolean onOptionsItemSelected(@NonNull MenuItem item) {  
 return super.onOptionsItemSelected(item);  
}

를 추가한다.