

A cluster of overlapping, semi-transparent geometric shapes in shades of blue, green, and red, located in the top-left corner of the slide.

# Android Programming

## Layout

A cluster of overlapping, semi-transparent geometric shapes in shades of light gray, located in the bottom-right corner of the slide.

**Prawit Pimpisan**  
**Computer Science**  
**RERU**

# Layout

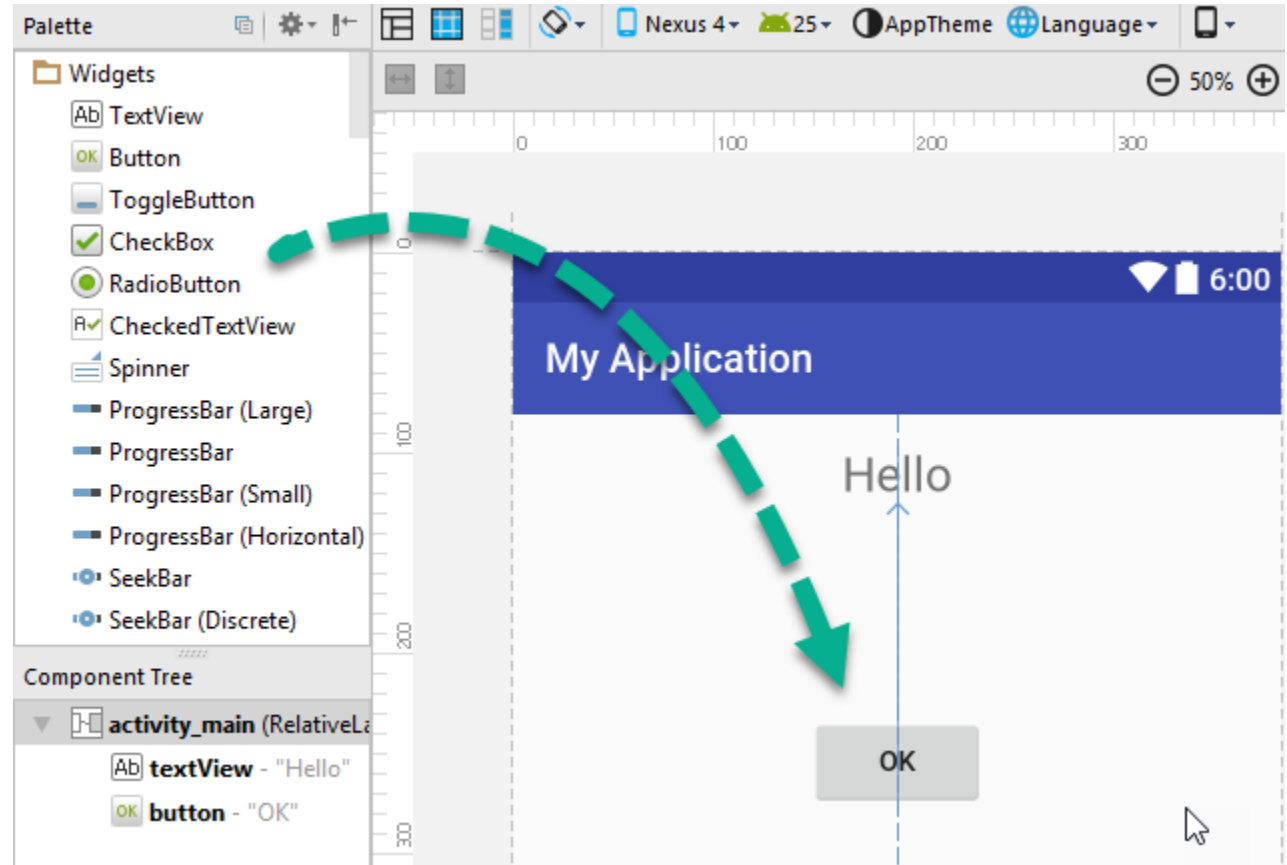
- Layout เป็นการออกแบบการจัดวางตำแหน่งให้กับส่วนต่างๆ ในหน้า view
- การออกแบบที่นิยมกับพัฒนา Android จะออกแบบเป็น view ในส่วนของโฟลเดอร์ res ที่ใช้ภาษา xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.dell.myapplication.MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!" />
</RelativeLayout>
```

# การออกแบบ Layout

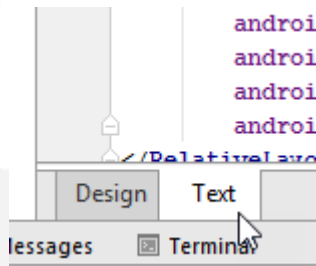
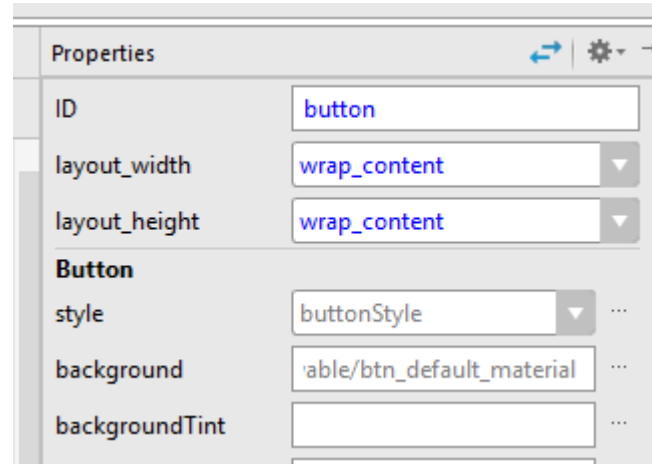
การออกแบบ  
โดยทั่วไปทำได้  
โดยการลาก Widget ที่ต้องการลงที่  
หน้า Layout ได้  
เลย



# การตั้งค่า Property

สามารถทำได้ 2 วิธีคือ

1. ตั้งค่าผ่าน Property tab โดยตรง
2. ตั้งค่าผ่าน XML



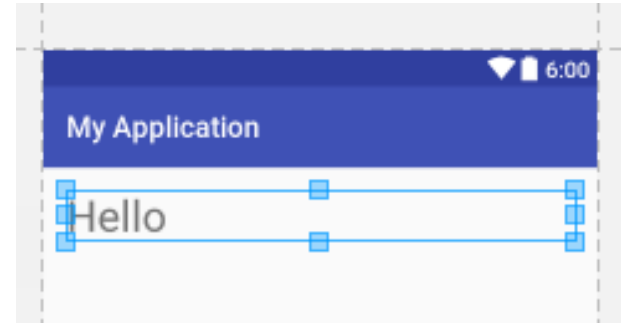
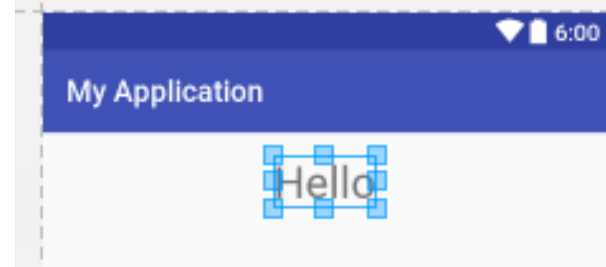
```
<TextView
    android:text="Hello"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/textView"
    android:layout_centerHorizontal="true"
    android:textSize="24sp"
    android:lineSpacingExtra="12sp" />
```

# หน่วยของขนาด

- dp : หน่วยที่ไม่ขึ้นกับความหนาแน่นของหน้าจอจริง แต่เป็นหน่วยเทียบเท่า ซึ่งถ้านำไปใช้กับอุปกรณ์ที่มีขนาดหน้าจอต่างแล้ว จะมีการปรับสัดส่วนขนาดให้เหมาะสมตามขนาดหน้าจอ
- sp : ลักษณะคล้ายกับ dp แต่แนะนำเฉพาะกับตัวอักษร เนื่องจากมีผลเมื่อผู้ใช้ตั้งค่า User's preference
- px : ขนาด pixel จริง แต่ไม่เหมาะกับการใช้งาน android เท่าไรนัก เนื่องจากอุปกรณ์ android มีหน้าจอหลากหลายขนาด
- pt : ขนาด  $1/72$  นิ้วบนอุปกรณ์จริง
- mm : ขนาดมิลลิเมตร
- in : ขนาดนิ้ว

# การกำหนดความสัมพันธ์ให้กับ View

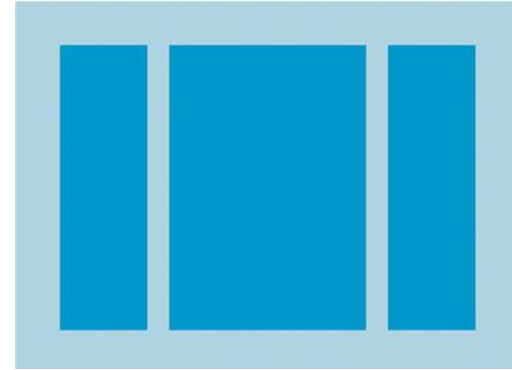
- `wrap_content` เป็นการกำหนดขนาดกรอบให้พอดีกับเนื้อหา-รายละเอียดของ Widget
- `fill_parent` กำหนดขนาดให้เต็มหน้าจอ หรือเต็มขนาดของ `layout` กำหนด
- `match_parent` เหมือนกับ `fill_parent` แต่เปลี่ยนมาใช้เป็น `match_parent` ตั้งแต่ `api8` ขึ้นไป



# การจัดเรียงองค์ประกอบของ layout

การจัดเรียง Layout ของ GUI ใน Android สามารถเลือกใช้รูปแบบการจัดได้ เช่น

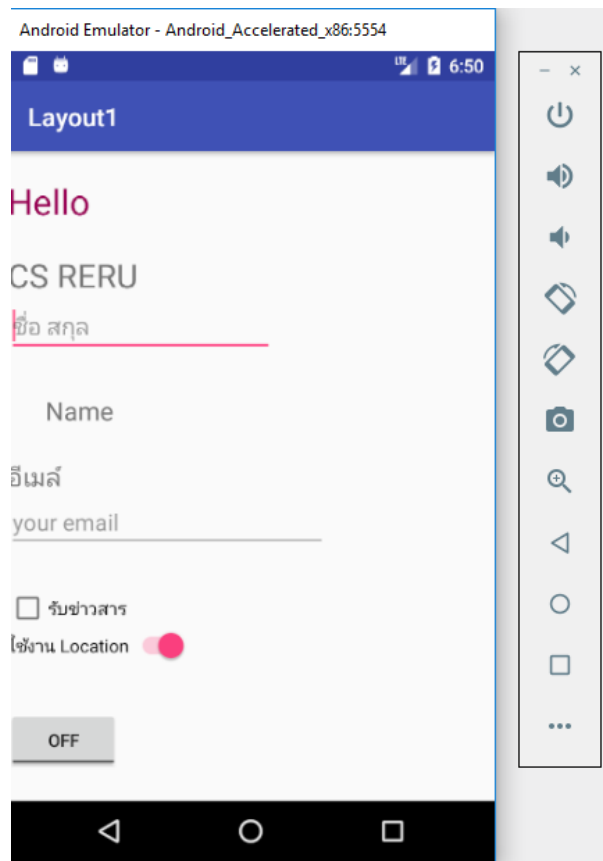
- Linear Layout เป็นการแบ่งการจัดเรียง layout ตามแนว โดยสามารถกำหนดได้ว่า จะแบ่งในแนวตั้งหรือแนวนอนผ่านคำสั่ง `android:orientation`
- Relative Layout เป็นการจัดวางตำแหน่งโดยอ้างอิงถึงองค์ประกอบอื่นๆ เช่น ต้องการวางปุ่มกด โดยกำหนดให้ปุ่ม อยู่ห่างจาก ขอบความ 20 dp
- Constrain Layout เน้นใช้เพื่อการลากวาง ช่วยให้พัฒนา Layout ได้ง่ายขึ้น ยืดหยุ่นมากขึ้น (Android Version ใหม่ ใช้ Constrain Layout ซึ่งจะมาแทนที่ Relative Layout)



# Linear layout

เป็นการจัดแบบเรียงกัน มี 2 แบบคือ แนวตั้งและ  
แนวนอน

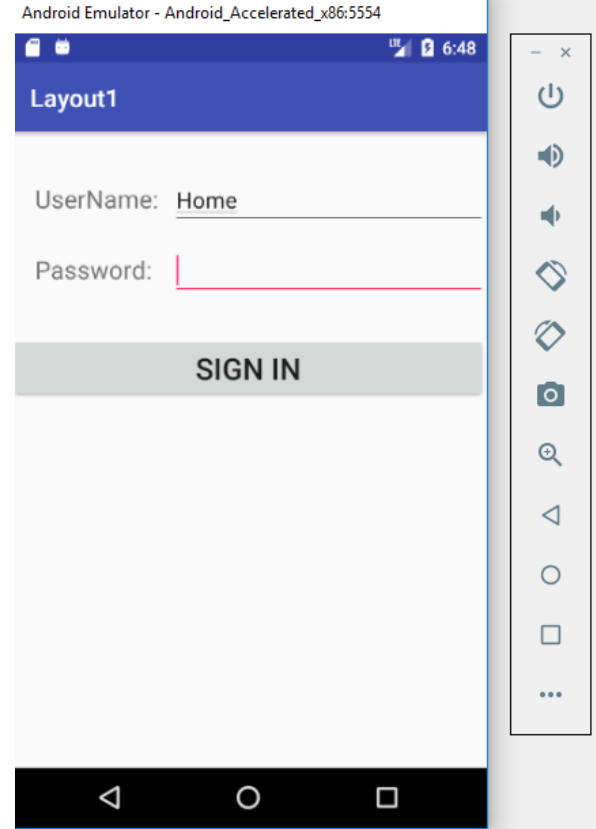
- `android:orientation="vertical"` แนวตั้ง
- `android:orientation="horizontal"` นแนวนอน





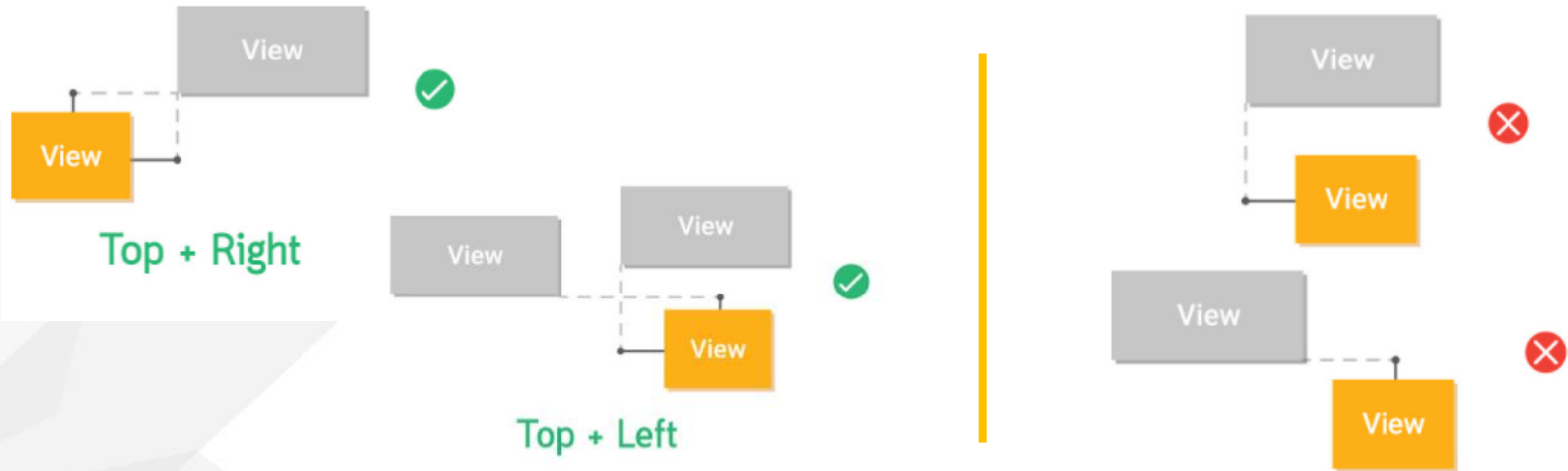
# Relative layout

เป็นการจัดโดยอ้างอิงจากวัตถุสิ่งอื่น  
เป็นการจัดที่สะดวก โดยการจับลากวางได้  
อย่างอิสระ แต่จะเริ่มมีปัญหาเมื่อมี widget  
ส่วนที่แสดงมากขึ้น ที่ทำให้เวลาลากวาง  
แล้ว ดันไม่ตรงตามตำแหน่งที่วาง



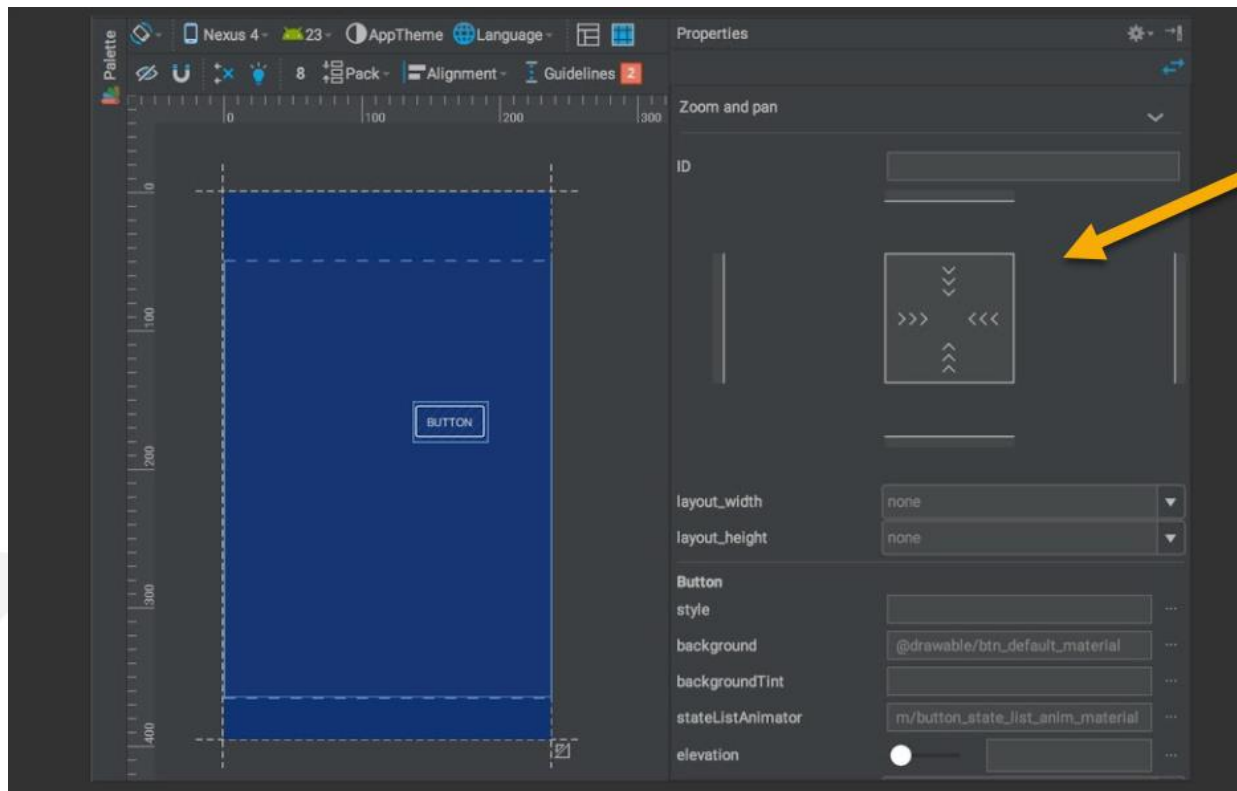
# Constraint layout

- มีการจัดรูปแบบคล้าย Relative Layout คืออ้างอิงสิ่งอื่นที่อยู่รอบตัวในทิศ Top, Bottom, Left, Right
- ต้องมีการเชื่อมต่อกับสิ่งอื่นในแนวดิ่ง (Top หรือ Bottom) อย่างน้อย 1 ด้าน
- ต้องมีการเชื่อมต่อกับสิ่งอื่นในแนวนอน (Left หรือ Right) อย่างน้อย 1 ด้าน



# Constraint layout

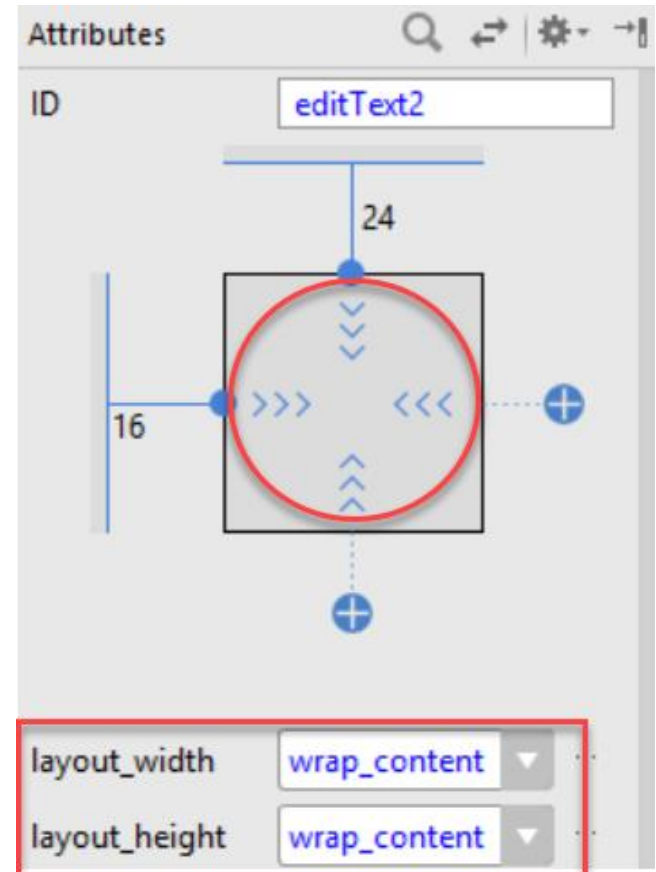
การออกแบบ View จะมี Properties เพิ่มขึ้นมา เพื่อใช้ตั้งค่าการออกแบบ Layout



# Constraint layout - Symbol

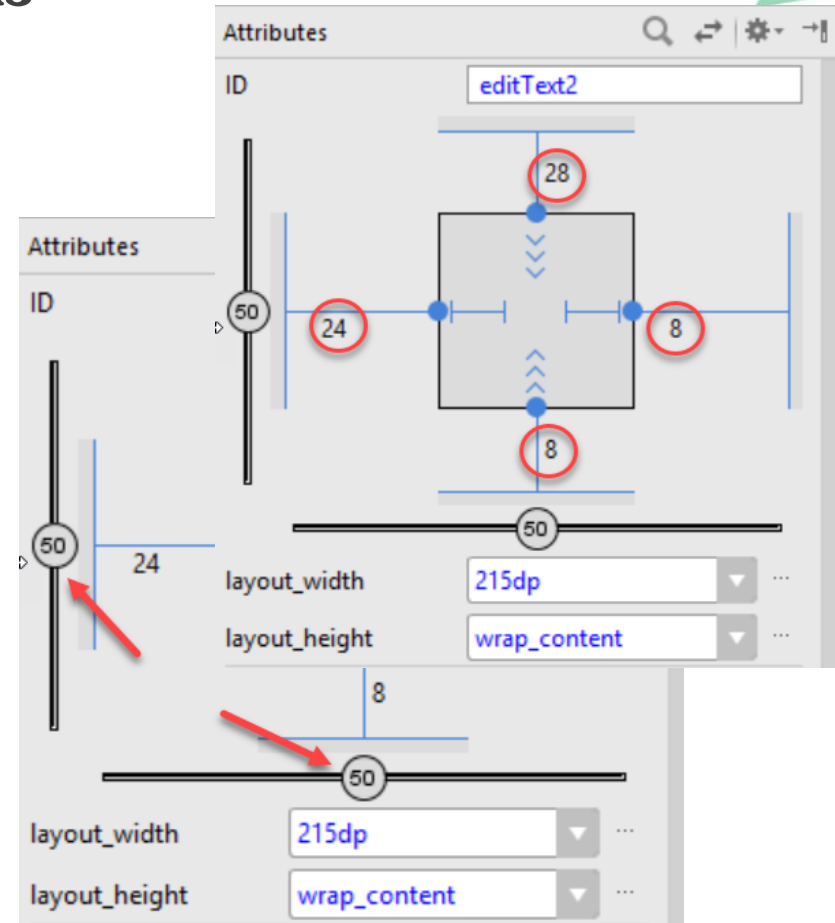
- ส่วนหลักที่ใช้ในการออกแบบคือ การตั้งค่า Properties ในการกำหนดระยะความห่าง, ความกว้าง และความสูง ของแต่ละ Widget
- ใช้สัญลักษณ์แทนการกำหนดขนาดด้วย wrap\_content, match\_parent และ dp

—	none or dp
>>>	wrap_content
	match_parent



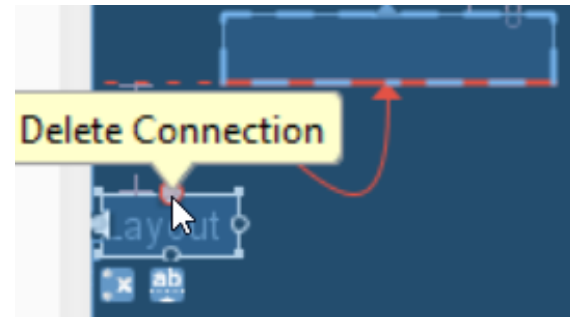
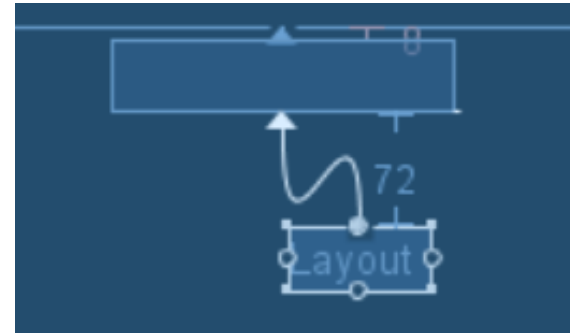
# Constraint layout – Margin, Bias

- ตัวเลขที่อยู่นอกกรอบหมายถึง ระยะห่างขอบ (Margin) ของทั้ง 4 ด้าน
- ตัวเลขแนวตั้งและแนวนอนคือค่า Bias เป็นค่าที่ใช้รับความเยื้อง (ค่าเริ่มต้นคือ 50-50 ที่ทำให้วัตถุอยู่ตำแหน่งตรงกลาง)



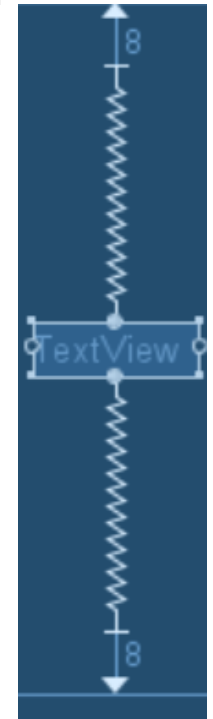
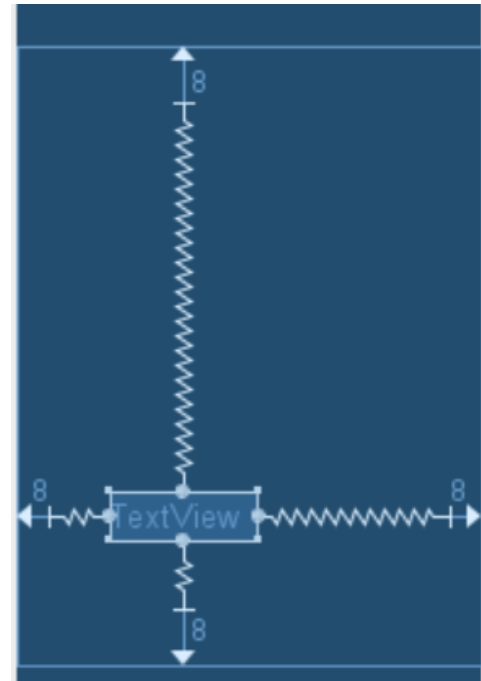
# Constraint layout – Constraint Line

- แต่ละวัตถุจะมีมุมกรอบที่สามารถลากปรับขนาดได้ทั้ง 4 ด้านตามต้องการ แล้วจะได้หน่วยเป็น dp ทันที
- จุดวงกลมตรงกลางใช้กำหนดด้านของแขนที่เชื่อมต่อกับวัตถุอื่น
- สามารถยกเลิกแขนเชื่อมต่อโดยการคลิกที่จุดวงกลมอีกครั้ง
- เส้นแขนเชื่อมต่อ เรียกว่า เส้น Constraint



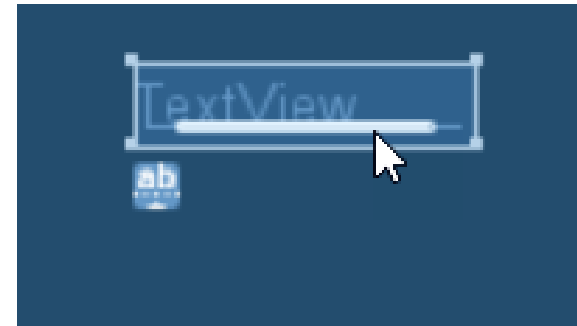
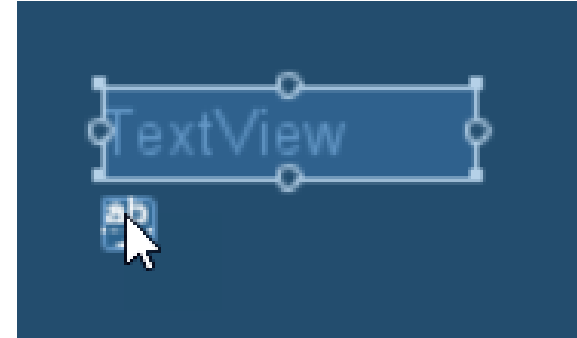
# Constraint layout - Position

- การสร้างเส้น **Constrain** จะต้องเชื่อมทั้งแนวตั้งและแนวนอน (อย่างน้อยด้านละ 1 ข้างเสมอ)
- ถ้าทำการเชื่อมทั้งด้านซ้ายและขวา วัตถุจะอยู่ตำแหน่ง **Center Horizontal**
- ถ้าทำการเชื่อมทั้งด้านบนและล่าง วัตถุจะอยู่ตำแหน่ง **Center Vertical**
- สามารถปรับตำแหน่งได้ โดยการคลิกลากที่วัตถุ เพื่อปรับค่า **Bias**



# Constraint layout - Baseline

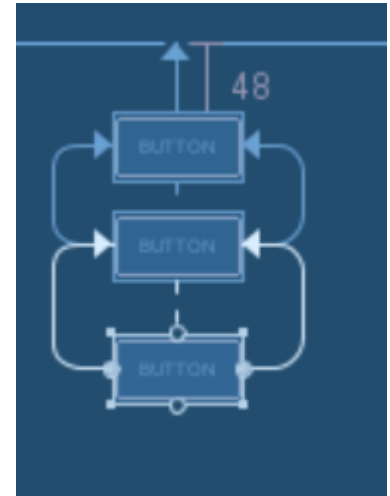
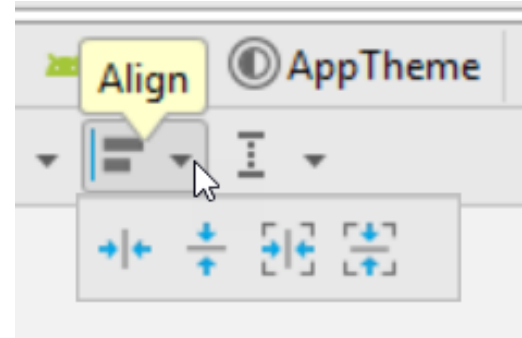
- การจัดตำแหน่งอ้างอิงนอกจากวิธีการปรับที่ 4 ด้านแล้ว ยังมีอีกวิธีคือ การอ้างอิงด้วย Baseline
- แต่ละวัตถุสามารถเลือกปรับ Baseline โดยการคลิกปุ่มเล็กกลางซ้าย แล้วจะปรากฏแถบ Baseline
- Baseline จะได้ระดับในแนวนอน จะต้องเพิ่มการอ้างอิงในแนวตั้งเพิ่มเติม เพื่อความสมบูรณ์ของตำแหน่ง
- Baseline จะต้องใช้อ้างอิงกับ Baseline ด้วยกันเท่านั้น จะอ้างอิงด้านทั้ง 4 ไม่ได้





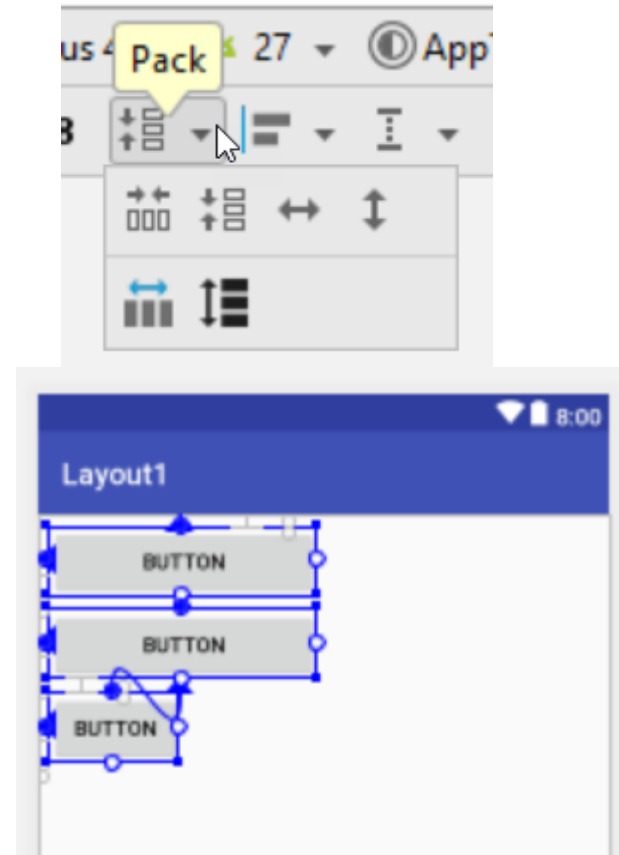
# Constraint layout - Alignment

- สามารถใช้เครื่องมือช่วยในการจัดแนวได้
- คลิกเลือกวัตถุที่ต้องการจัดตำแหน่งพร้อมกัน แล้วเลือกรูปแบบที่ต้องการจัด



# Constraint layout - Linear

- สามารถใช้เครื่องมือช่วยจัดในรูปแบบของ Linear ได้ทั้งแนวตั้งและแนวนอน

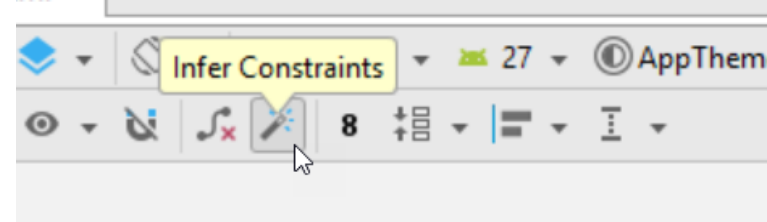


# Constraint layout – Infer Constraints

- เป็นการสร้างเส้น Constraint ให้โดยอัตโนมัติ
- ออกแบบตำแหน่งต่างๆของวัตถุให้เรียบร้อย โดยที่ไม่ต้องเปิด Auto Connect Constraints
- เลือกวัตถุทั้งหมด จากนั้นกดปุ่ม Infer Constraints
- การใช้งานอาจจะยังไม่ได้ผลลัพธ์ 100 % แต่ได้ความสะดวกรวดเร็วในการออกแบบ



ปิด Auto Constraint



# Constraint layout



จงออกแบบหน้า Layout โดยใช้ตัวอย่างจาก Linear Layout และ Relative Layout