## 2.2 Linear, Relative, and Constraint layouts

#### LinearLayout / android:orientation

The LinearLayout arranges child Views horizontally by default or vertically if the XML attribute android:orientation is "vertical".

```
Figure 2.2.1: LinearLayout with horizontal (left) and vertical (right)
orientation.
<LinearLayout</pre>
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout width="match parent"
    android:layout height="match parent"
    android:orientation="horizontal">
    <Button
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="Button 1" />
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="Button 2" />
    <Button
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="Button 3" />
</LinearLayout>
                                               LinearLayout Example
    LinearLayout Example
Button 1
          Button 2
                   Button 3
                                            Button 1
                                           Button 2
                                           Button 3
                                                                              Feedback?
```

# RelativeLayout

The RelativeLayout is a ViewGroup that arranges child Views in relation to each other.

Figure 2.2.2: RelativeLayout example with a TextView and four checkbox widgets.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"</pre>
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity main"
    android:layout_width="match_parent"
    android:layout height="match parent"
    android:padding="6dp">
    <TextView
        android:id="@+id/question text view"
        android:layout width="wrap content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout alignParentTop="true"
        android:text="Which languages are you fluent in?" />
    <CheckBox
        android:id="@+id/english check box"
        android:text="English"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout below="@id/question text view" />
    <CheckBox
        android:id="@+id/spanish_check_box"
        android:text="Spanish"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout_below="@id/english_check_box" />
    <CheckBox
        android:id="@+id/chinese_check_box"
        android:text="Chinese"
        android:layout marginLeft="30dp"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:layout below="@id/question text view"
        android:layout_toRightOf="@id/english_check_box"/>
    <CheckBox
        android:id="@+id/other check box"
        android:text="Other"
        android:layout width="wrap content"
        android:layout_height="wrap_content"
        android:layout below="@id/chinese check box"
        android:layout alignLeft="@id/chinese check box" />
</RelativeLayout>
```

# RelativeLayout Example Which languages are you fluent in? English Chinese Spanish Other

Feedback?

## ConstraintLayout

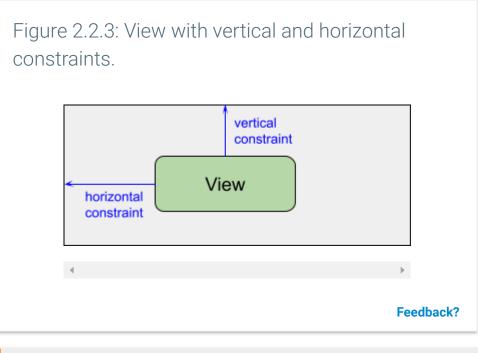
The ConstraintLayout is a ViewGroup that arranges child Views in relation to each other using horizontal and vertical constraints.

#### horizontal constraint

A horizontal constraint aligns a View's left or right side to the left or right side of another View or ViewGroup.

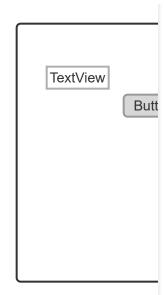
#### vertical constraint

A vertical constraint aligns a View's top or bottom sides to the top or bottom side of another View or ViewGroup.





```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout</pre>
   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">
    <TextView
        android:id="@+id/text view1"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="TextView"
        app:layout constraintLeft toLeftOf="parent"
        app:layout constraintTop toTopOf="parent"
        android:layout marginLeft="20dp"
        android:layout marginTop="40dp" />
    <Button
        android:id="@+id/button1"
        android:layout width="wrap content"
        android:layout height="wrap content"
        android:text="Button"
        app:layout constraintLeft toRightOf="@id/text view1"
        app:layout constraintTop toTopOf="@id/text view1"
        android:layout_marginLeft="10dp"
        android:layout marginTop="20dp" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



#### Captions ^

- 1. The ConstraintLayout has a TextView child with constraints that align the TextView's left edge with the parent layout's left edge and the top edge with the parent layout's top edge.
- 2. The Button child's left constraint aligns the Button's left edge to the TextView's right edge, and the top constraint aligns the top edge to the TextView's top edge.
- 3. Adding a left margin moves the TextView 20dp from the layout's left edge, and adding a top margin moves the TextView 40dp from the layout's top edge.
- 4. Adding margins to the Button moves the Button 10dp to the right and 20dp below the Button's constrained location.

Feedback?

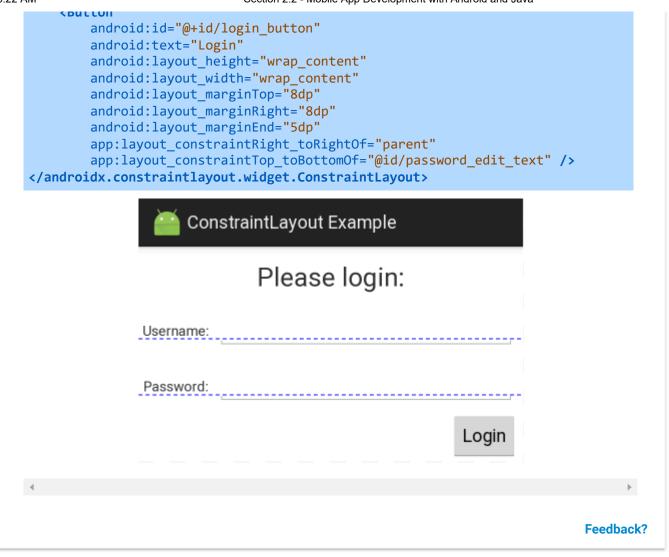
#### baseline constraint

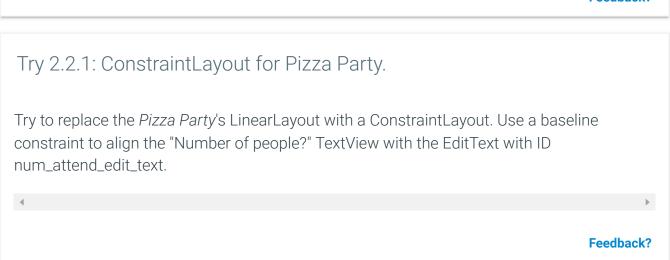
A baseline constraint is a vertical constraint that aligns the text baselines of two Views.

Figure 2.2.4: ConstraintLayout using baseline constraints.

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout</pre>
```

```
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">
<TextView
    android:id="@+id/login_text_view"
   android:text="Please login:'
    android:textSize="26sp"
   android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    app:layout constraintBottom toTopOf="@id/username edit text"
   app:layout constraintRight toRightOf="parent"
   android:layout marginTop="4dp"
   android:layout_marginBottom="8dp"
   app:layout constraintLeft toLeftOf="parent"
    app:layout constraintTop toTopOf="parent" />
<TextView
    android:id="@+id/username text view"
    android:text="Username:"
    android:layout_height="wrap_content"
   android:layout_width="wrap_content"
   android:layout marginTop="30dp"
   android:layout marginLeft="4dp"
    app:layout constraintTop toBottomOf="@id/login text view"
    app:layout_constraintLeft_toLeftOf="parent" />
<EditText
    android:id="@+id/username_edit_text"
    android:layout_width="0dp'
    android:layout height="wrap content"
   android:layout marginLeft="8dp"
    android:layout marginRight="8dp"
    app:layout_constraintRight_toRightOf="parent"
    app:layout constraintLeft toRightOf="@id/username text view"
    app:layout constraintBaseline toBaselineOf="@id/username text view" />
<TextView
    android:id="@+id/password text view"
   android:text="Password:"
   android:layout height="wrap content"
    android:layout_width="wrap_content"
    android:layout marginLeft="5dp"
    android:layout_marginTop="30dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@id/username_edit_text" />
<EditText
    android:id="@+id/password_edit_text"
    android:layout_width="0dp"
    android:layout height="wrap content"
   android:inputType="textPassword"
   android:layout_marginRight="8dp"
    app:layout constraintRight toRightOf="parent"
    app:layout constraintLeft toLeftOf="@id/username edit text"
    app:layout constraintBaseline toBaselineOf="@id/password text view" />
```





This section does not contain presentation elements.

How was this Provide feedback section?