

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".

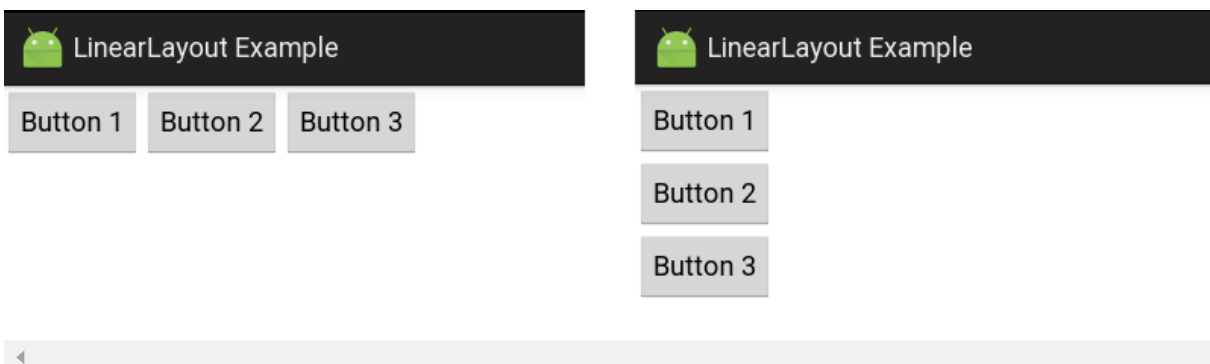
Figure 2.2.1: LinearLayout with horizontal (left) and vertical (right) orientation.

```
<LinearLayout
    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" />

    <Button
        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>
```



[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"
    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

ConstraintLayout

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

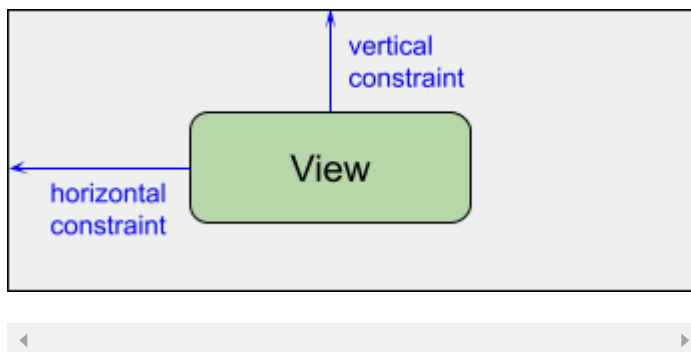
horizontal constraint

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.

Figure 2.2.3: View with vertical and horizontal constraints.

[Feedback?](#)

PARTICIPATION ACTIVITY

2.2.4: ConstraintLayout example.

[Start](#)

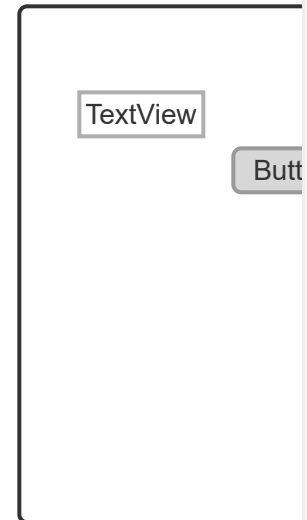
2x speed

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

    <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
```

```

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" />

```

```
<Button
    android:id="@+id/login_button"
    android:text="Login"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:layout_marginTop="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginEnd="5dp"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/password_edit_text" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



ConstraintLayout Example

Please login:

Username:

Password:

Login

[Feedback?](#)

Try 2.2.1: ConstraintLayout for Pizza Party.

Try to replace the *Pizza Party*'s *LinearLayout* with a *ConstraintLayout*. Use a baseline constraint to align the "Number of people?" *TextView* with the *EditText* with ID `num_attend_edit_text`.

[Feedback?](#)

This section does not contain presentation elements.

How
was this
section?



[Provide feedback](#)