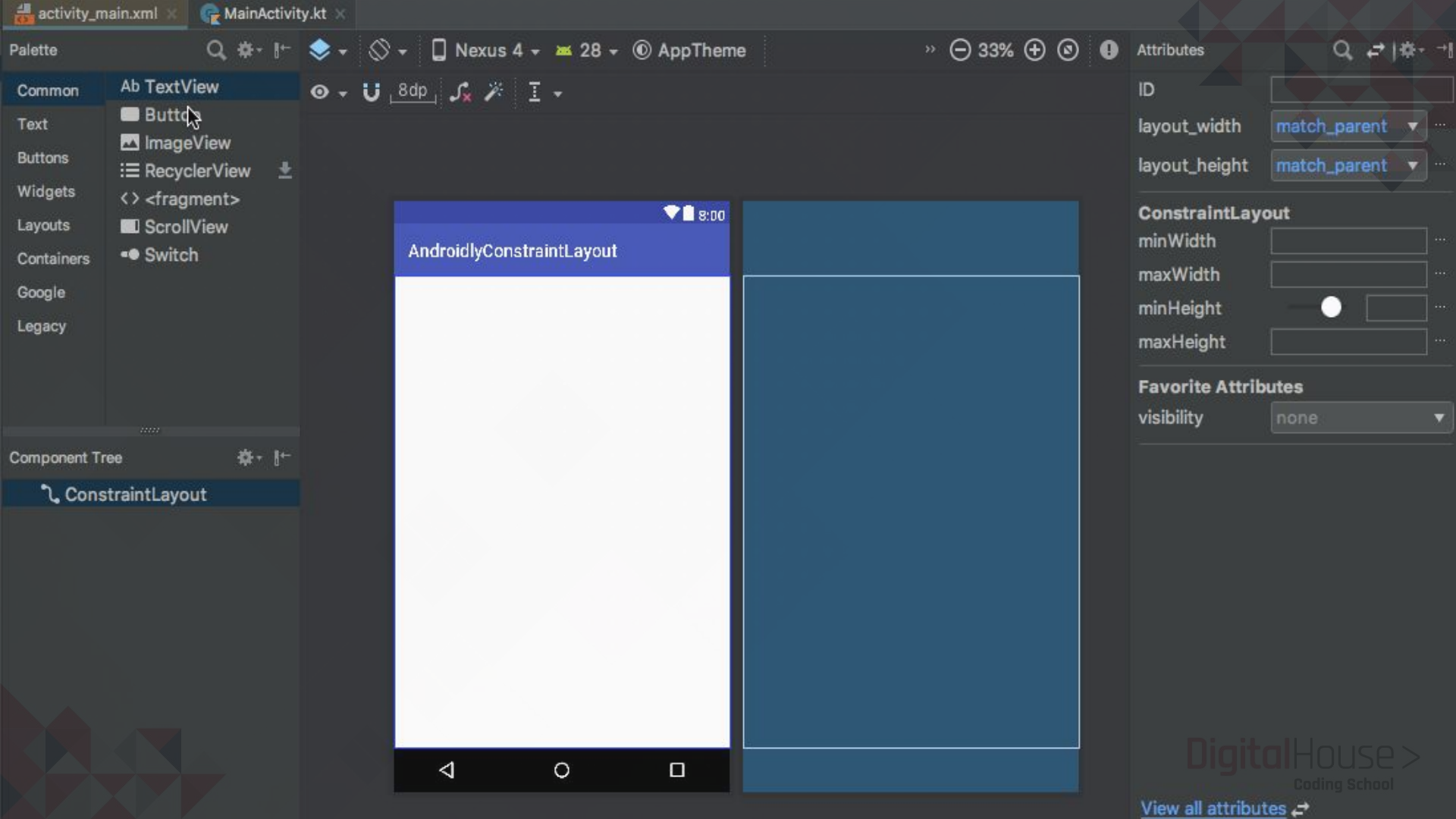
Aula 24 - Layouts

DigitalHouse >
Coding School

Layouts

DigitalHouse >
Coding School

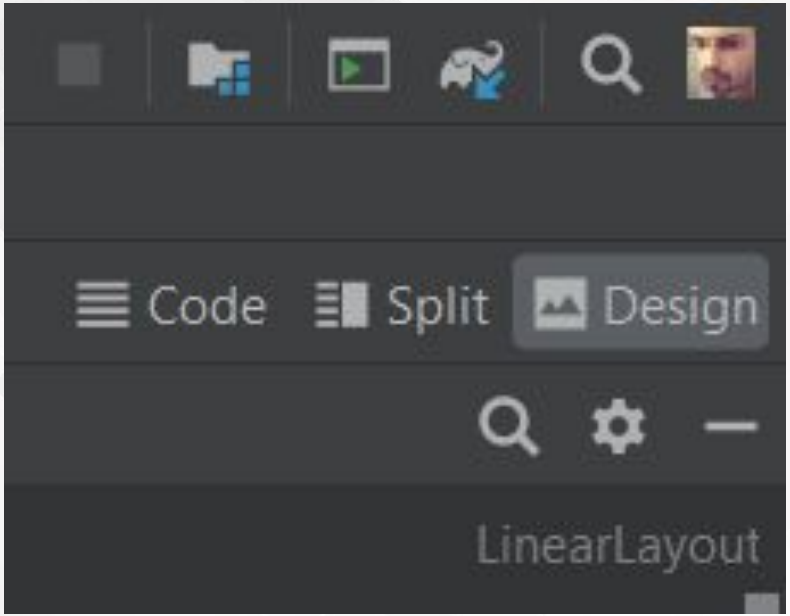

activity_main.xml
MainActivity.kt
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Ab TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Component Tree
ConstraintLayout
Nexus 4
28
AppTheme
33%
8dp
8:00
AndroidlyConstraintLayout
Attributes
ID
layout_width
match_parent
layout_height
match_parent
ConstraintLayout
minWidth
maxWidth
minHeight
maxHeight
Favorite Attributes
visibility
none
View all attributes
DigitalHouse
Coding School

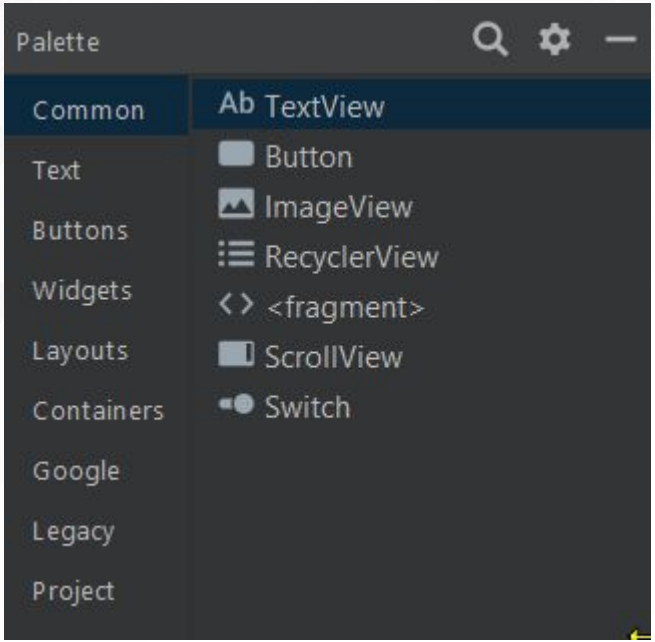
Xtensible Markup Language

Usada para montar uma receita de uma tela e prever seu comportamento.

```
<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:background="@color/blue_600"
    android:gravity="center"
    android:orientation="vertical"
    tools:ignore="HardcodedText"
    >

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Um Texto"
        />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Um Botão"
        />

    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Um Campo de Edição de Texto"
        />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:contentDescription="Uma imagem"
        />

</LinearLayout>
```

Menus importantes:





DigitalHouse > Coding School

exemplo 〉 app 〉 src 〉 main 〉 res 〉 layout 〉 act_exemplo

exemplo-app | Pixel 3 API 29

act_exemplo

Code | Split | Design

Palette

Common
Text
Buttons
Widgets
Layouts
Containers
Google
Legacy
Project

CircleImageView
BowserView
VerticalRecycler
VerticalGridRecycler
PeterWebView

Pixel | 30 | NoActionBar | Default (en-us)

Component Tree

LinearLayout (vertical)
textView "Um Texto"
button "Um Botão"
edittext
imagem

Attributes

<unnamed> LinearLayout

id

Declared Attributes

Layout

layout_width: match_parent
layout_height: match_parent
visibility:
visibility:

Common Attributes

orientation: vertical
gravity: center
alpha:

All Attributes

accessibilityLiveRegion
accessibilityTraversalAfter
accessibilityTraversalBefo...
actionBarNavMode
addStatesFromChildren
alpha
alwaysDrawnWithCache
animateLayoutChanges
animationCache
background: @color/blue_600
backgroundTint
backgroundTintMode
baselineAligned
baselineAlignedChildIndex
clickable
clipChildren
clipToPadding

1:1

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Um Texto"
    />
```

Como comunicar entre XML e Kotlin?

```
<Button
    android:id="@+id/um_botao"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Um Botão"
    />
```

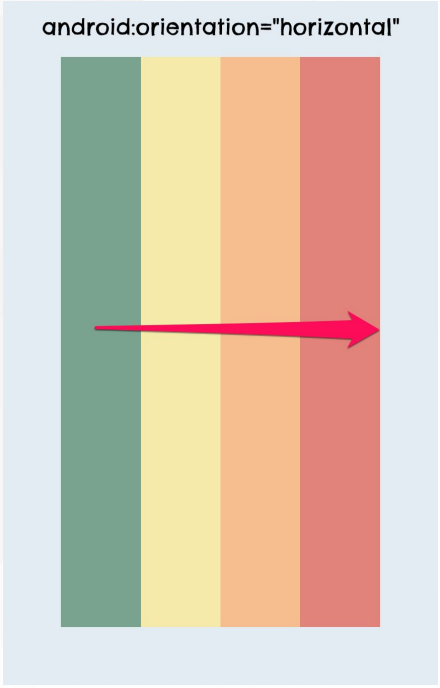
Como comunicar entre XML e Kotlin?

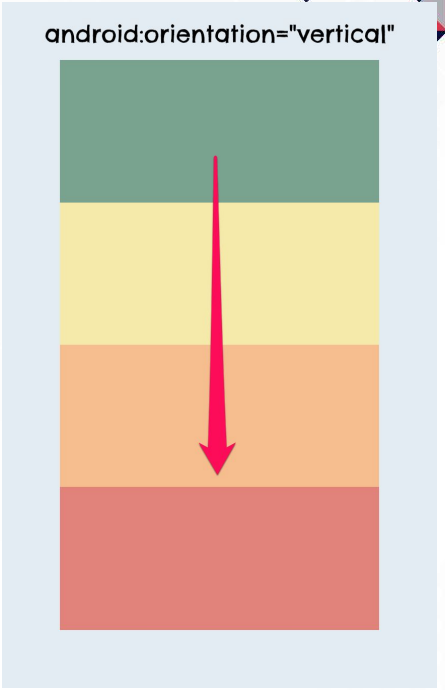
```
class ActExemplo : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.act_exemplo)
        val umBotao: Button = findViewById(R.id.um_botao)
    }

}
```

```kotlin
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    findViews()
}

private fun findViews() {
    setContentView(R.layout.act_exemplo)
    val botao: Button = findViewById(R.id.button)
    val texto: TextView = findViewById(R.id.textView)
    val edit: EditText = findViewById(R.id.edittext)
    val image: ImageView = findViewById(R.id.imageView)

    botao.setOnClickListener { it: View!
        println("Olá, Android!")
    }
}
```

LinearLayout

Alinha os elementos um após o outro, horizontal ou verticalmente





activity_main.xml

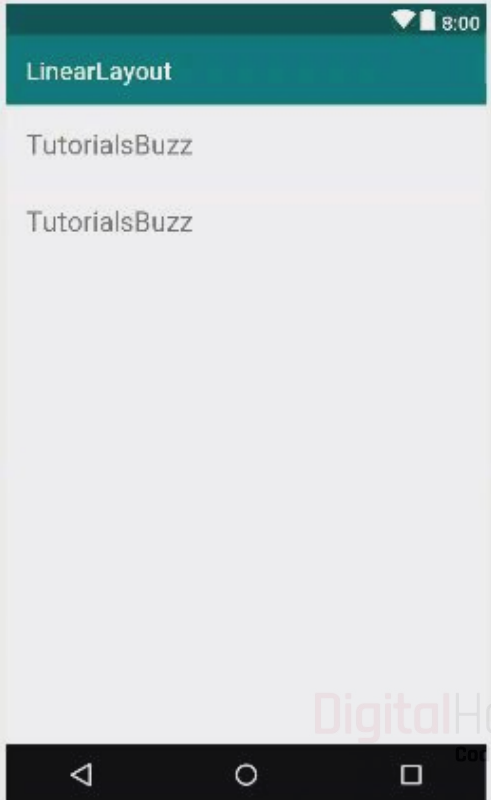
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/andr
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:text="@string/text_label"
        android:textSize="22sp" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:padding="16dp"
        android:text="@string/text_label"
        android:textSize="22sp" />

</LinearLayout>
```

Preview

Convert orientation to horizontal

Palette

Nexus 4 28 33%



Propriedades

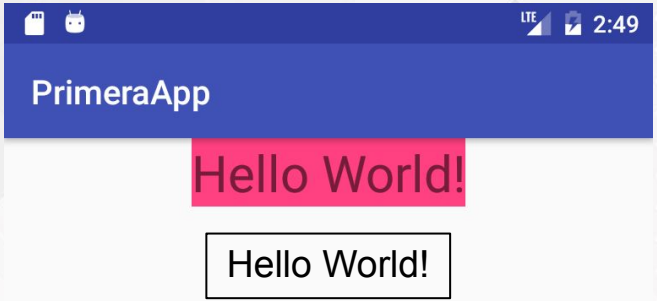
A maioria dos elementos possuem as mesmas propriedades:

- **Width**: Determina a largura
  `android:layout_width`
- **Height**: Determina a altura
  `android:layout_height`

Determinando largura e altura

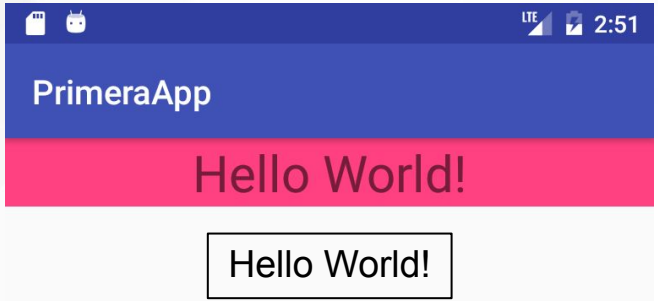
wrap_content

Determina que o tamanho da visualização seja igual ao seu conteúdo.



Match_parent

Determina que o tamanho da visualização será o mesmo que de seu pai.

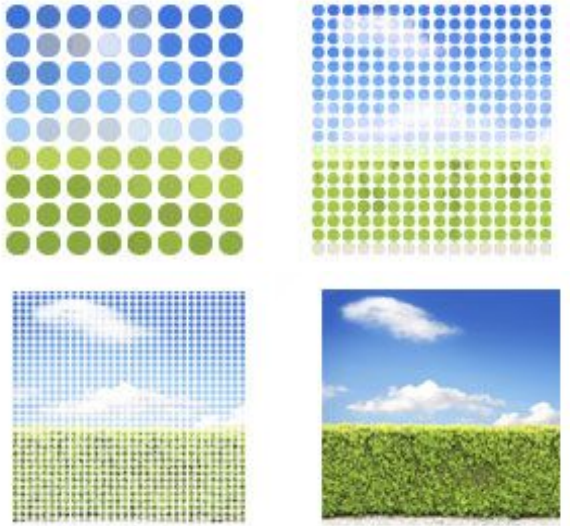


Determinando largura e altura

Caso queiramos que um elemento tenha um tamanho específico, um valor de **dp** (densidade de pixels) pode ser atribuído a ele.

Equivale a uma quantidade de pixels
de acordo com resolução do dispositivo

```
<TextView
android:layout_width="90dp"
android:layout_height="50dp"
android:text="@string/texto"
android:id="@+id/textView"/>
```

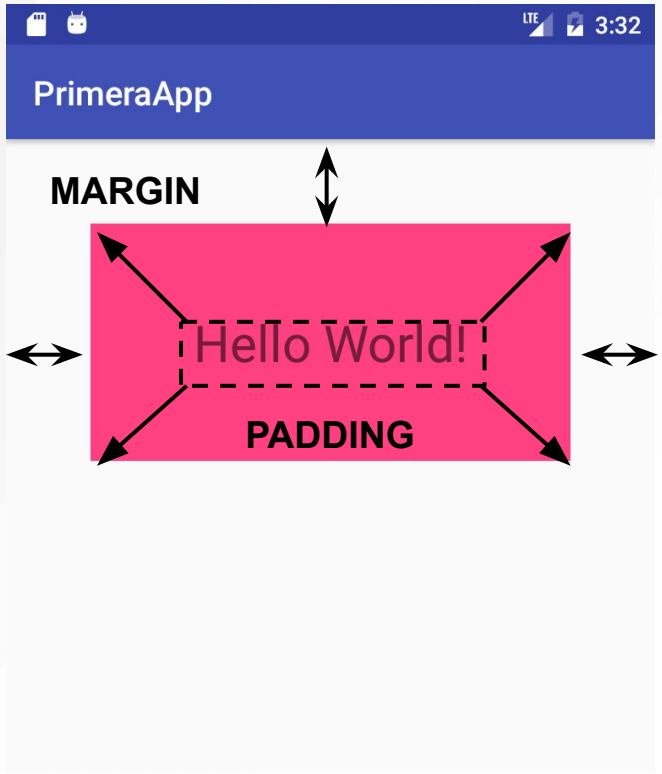


DigitalHouse > Coding School

Propriedades

Margin
Distância entre os elementos de um mesmo layout.
`android:layout_margin`

Padding
Distância entre a borda e o texto.
`android:padding`

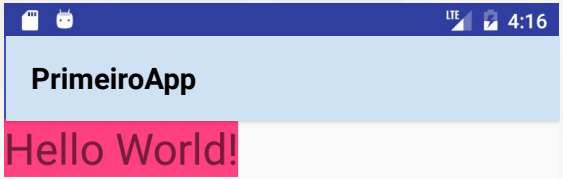


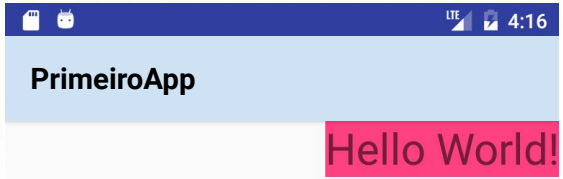
Propriedades

Gravity

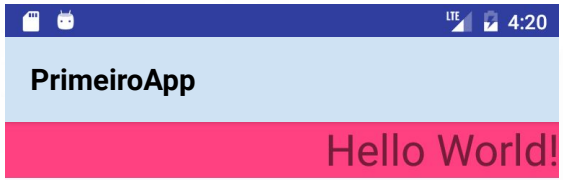
Determina a posição do conteúdo dentro de uma visualização.

`android:layout_gravity`





`android:gravity`



DigitalHouse > Coding School

DigitalHouse > Coding School