Desenvolvimento Para Dispositivos móveis

Prof. Me. Clênio Silva
E-mail: clenio.silva@uniube.br

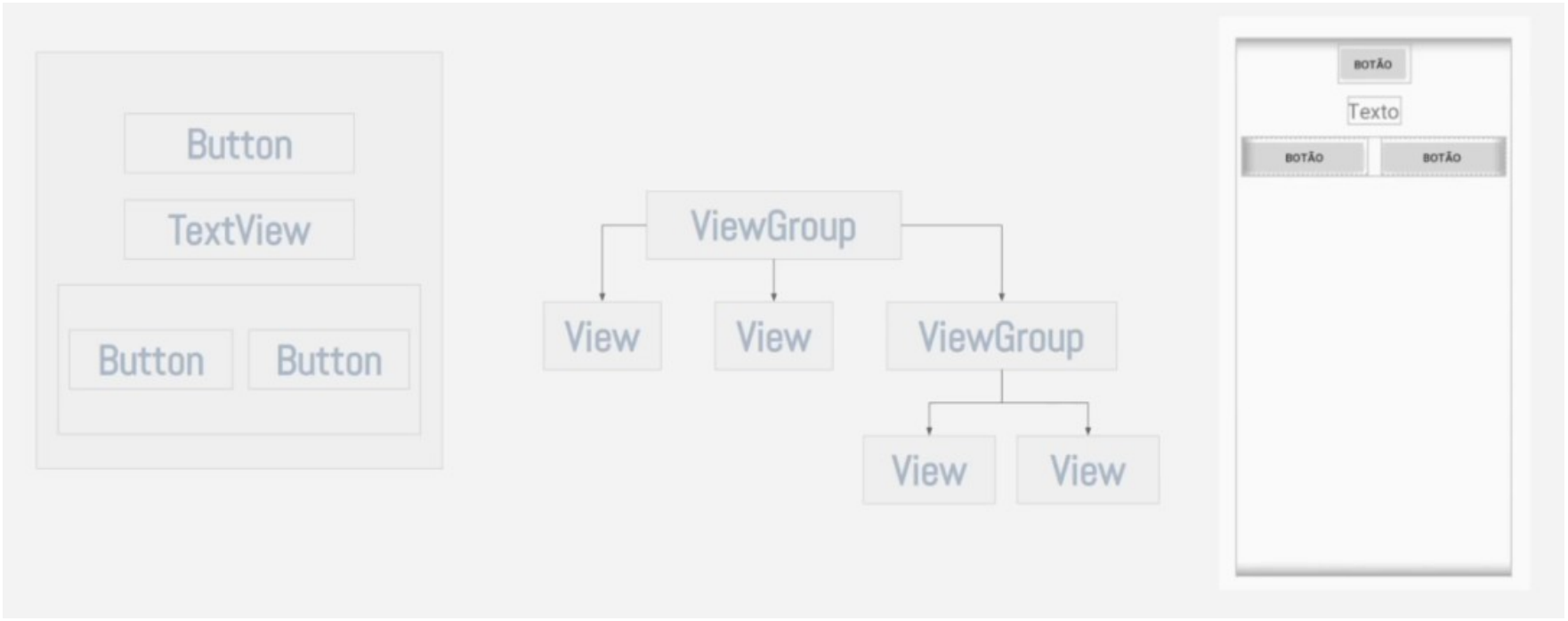
Android Layout - conceitos

- Define a estrutura da interface gráfica
  - Como os elementos se comportam e se relacionam
- Todos os elementos dentro do layout respeitam a hierarquia de View e ViewGroup

View	ViewGroup
Geralmente representam elementos na interface que o usuário possa interagir Button, EditText, Swutch, ImageView, CheckBox	Elementos de layout que definem o comportamento de outros elementos. Um agrupador de elementos de interface LinearLayout, RelativeLayout, ConstraintLayout, RecyclerView

- Um elemento de layout pode ser definido via código ou como XML

Hierarquia de layout



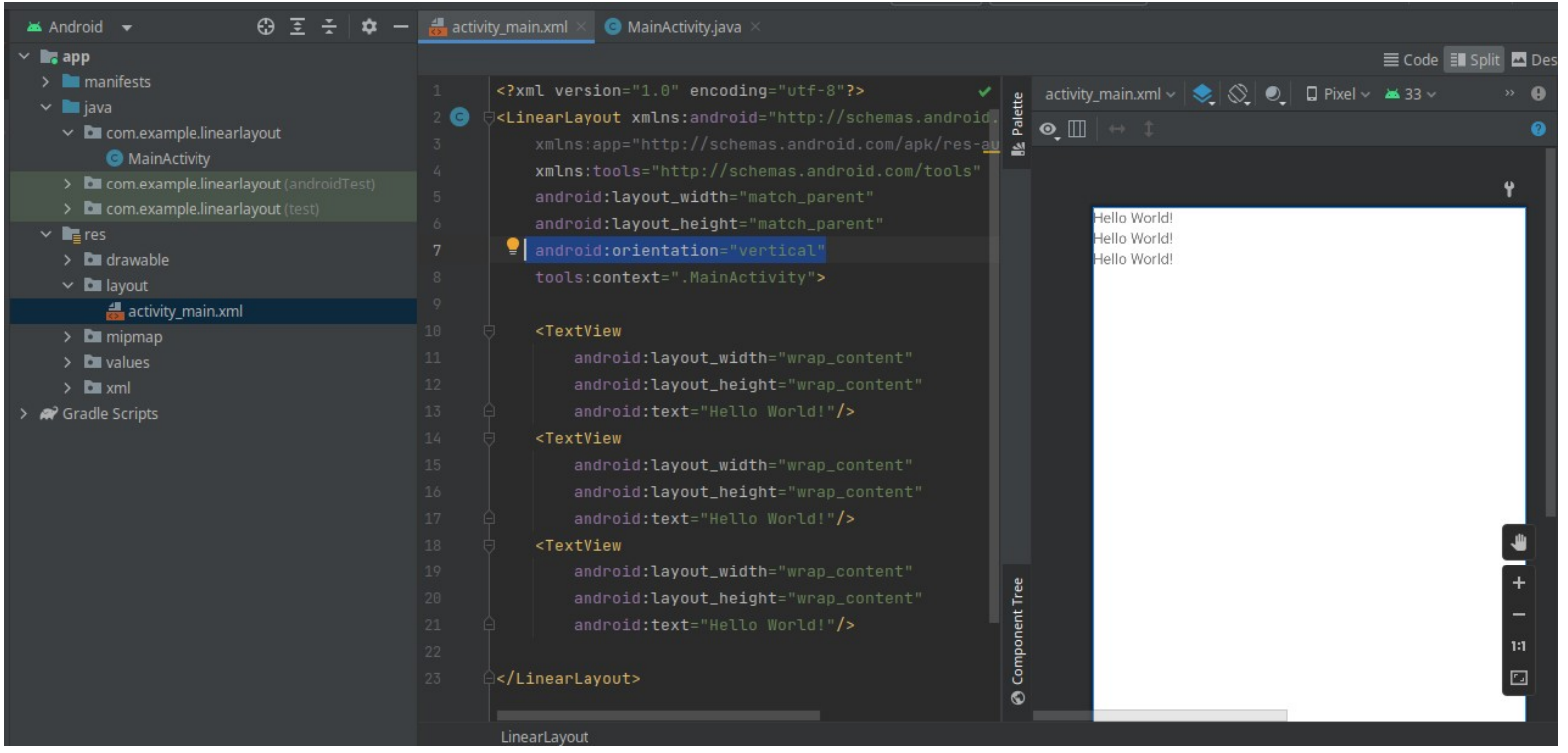
Boas práticas

- A quantidade de elementos na tela vão consumir mais processamento da aplicação
  - Deve-se criar layout simples e manter a hierarquia plana
- Usar as medidas recomendadas para largura e altura de elementos
  - Tamanho de fonte devem usar a media “sp” (Escala independente de pixels)
  - Tamanho de elementos, margins, paddings, etc devem usar “dp” (Densidade independente de pixels)
- Largura e altura também podem ser definidas relativas ao elemento pai ou próprio elemento

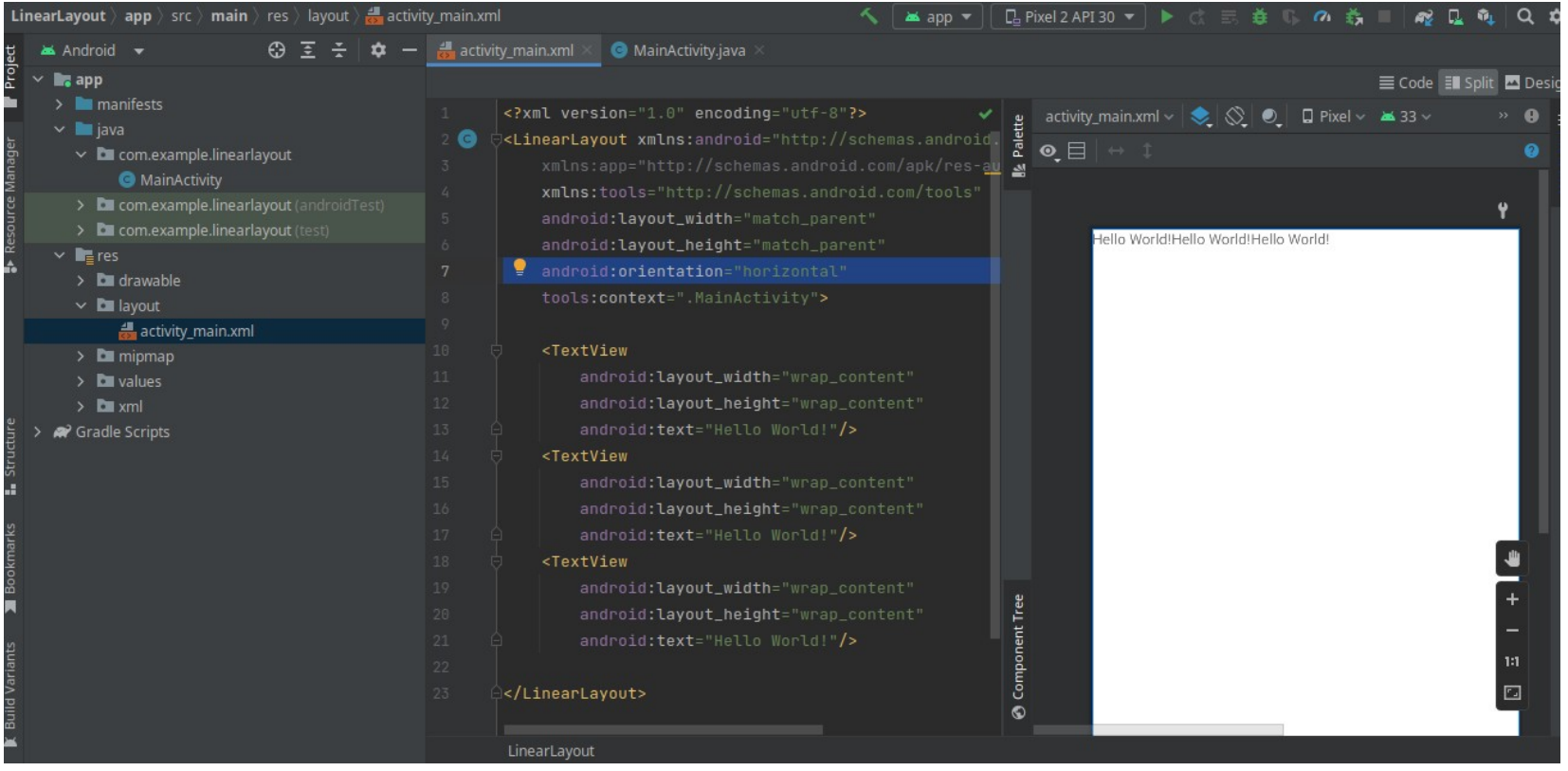
LinearLayout

- Crie um novo projeto:
  - Project Template = **Empty Activity** (Projeto em branco sem configurações)
  - Name = **LinearLAyout**
  - Language = **Java**

LinearLayout – definindo orientação - vertical

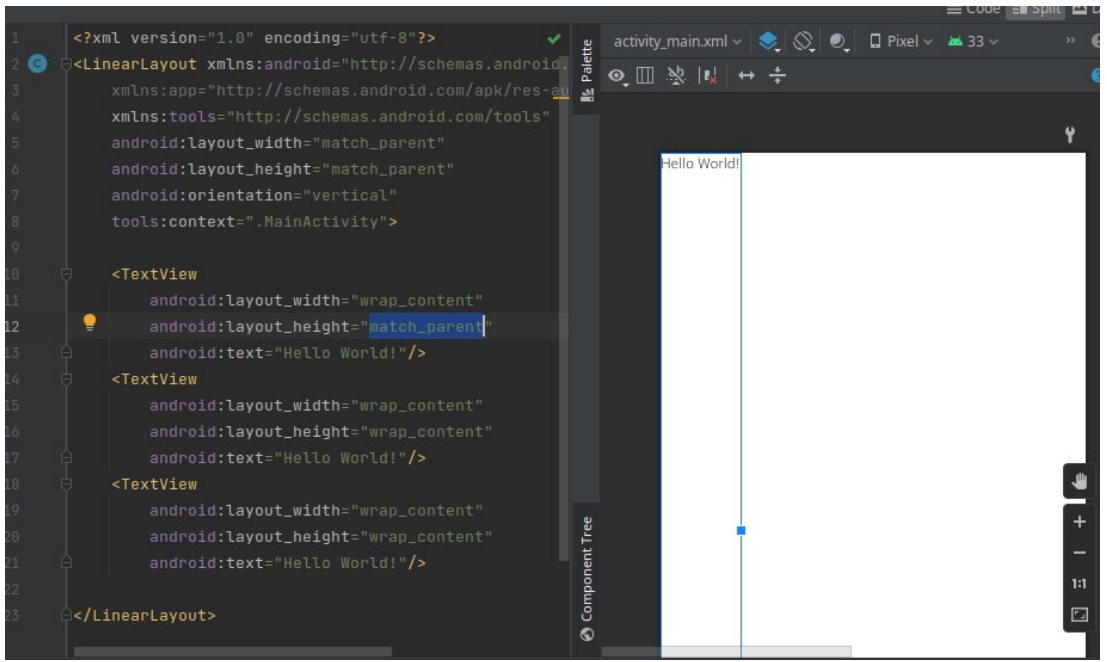


LinearLayout – definindo orientação - horizontal



LinearLayout – Alterando altura e largura dos elementos

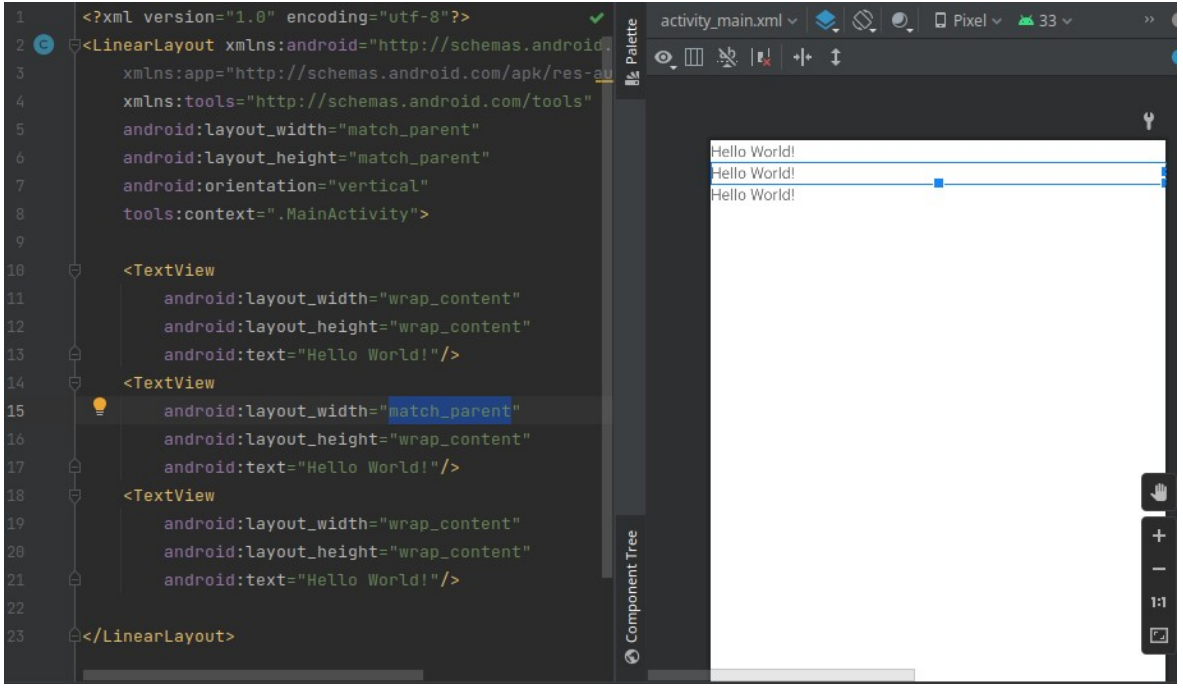
- match_parent: ocupa o tamanho do elemento pai
- wrap_content: ocupa o tamanho necessário



android:layout_width="wrap_content"
android:layout_height="match_parent"

LinearLayout – Alterando altura e largura dos elementos

- match_parent: ocupa o tamanho do elemento pai
- wrap_content: ocupa o tamanho necessário



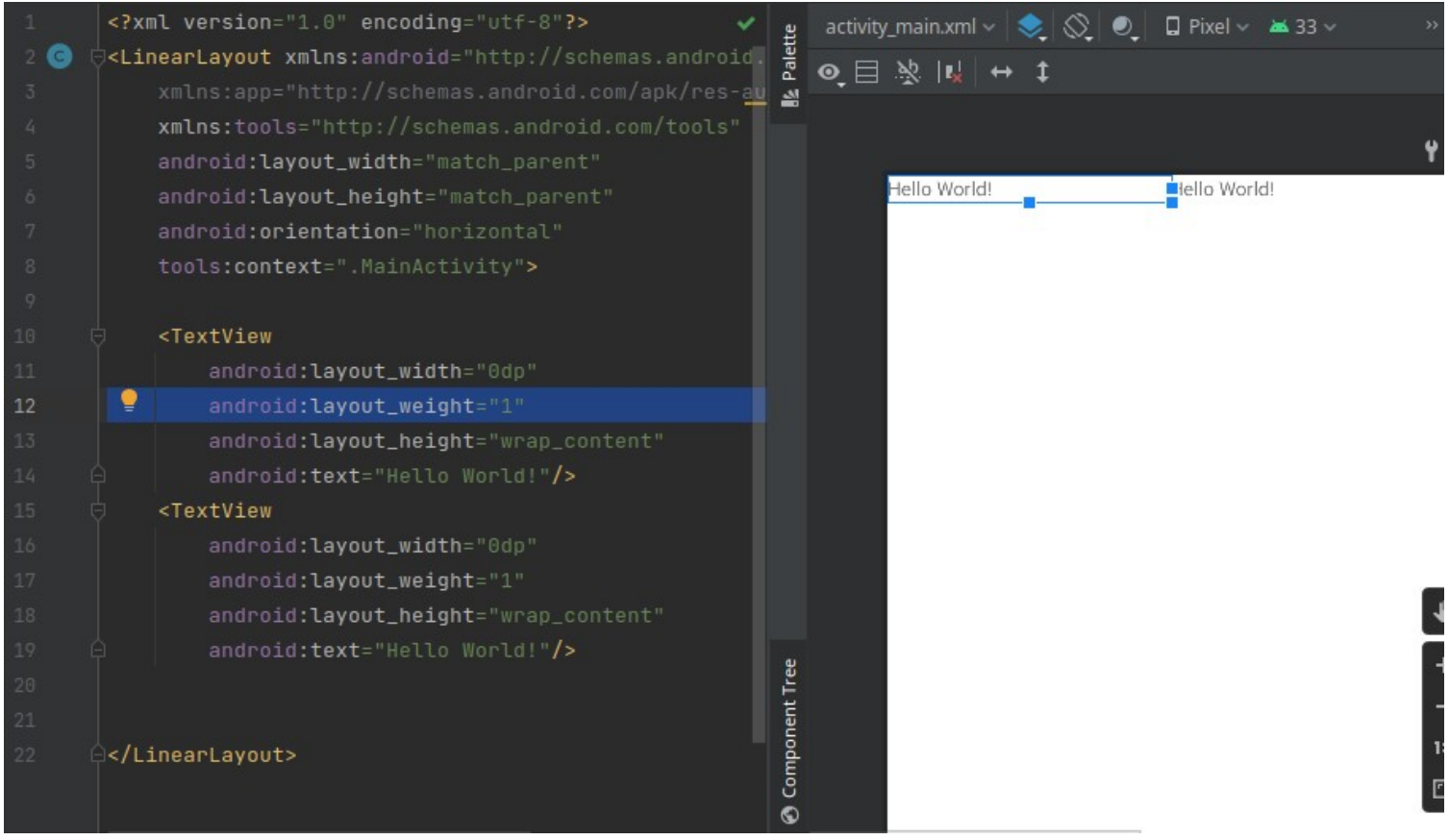

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android
    xmlns:app="http://schemas.android.com/apk/res-au
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"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

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

</LinearLayout>
```

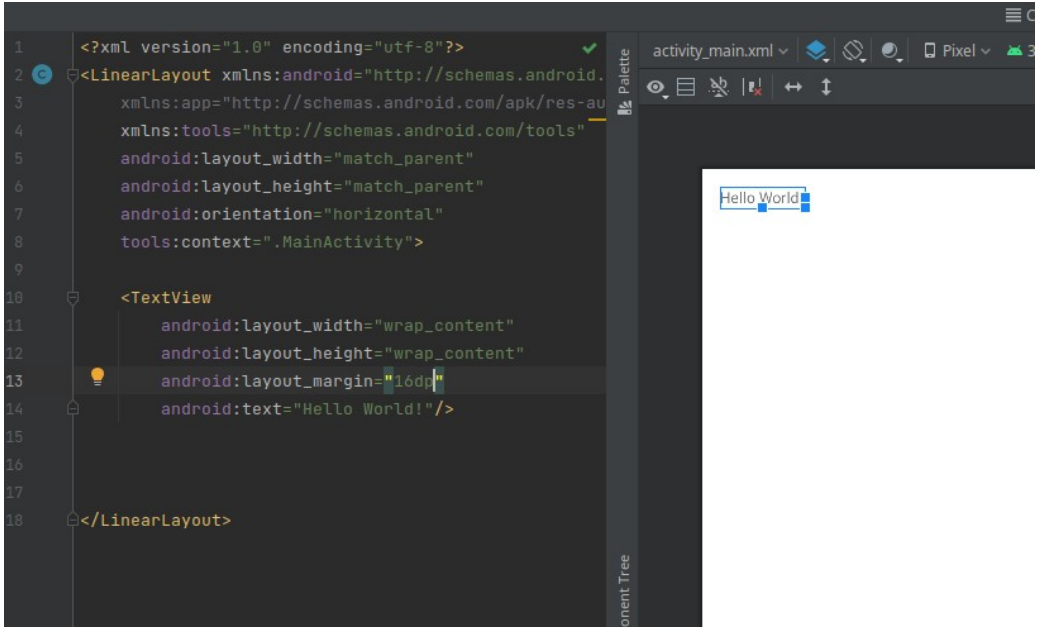
android:layout_width="match_parent"
android:layout_height="wrap_conten"

LinearLayout – Distribuindo elementos em um layout horizontal



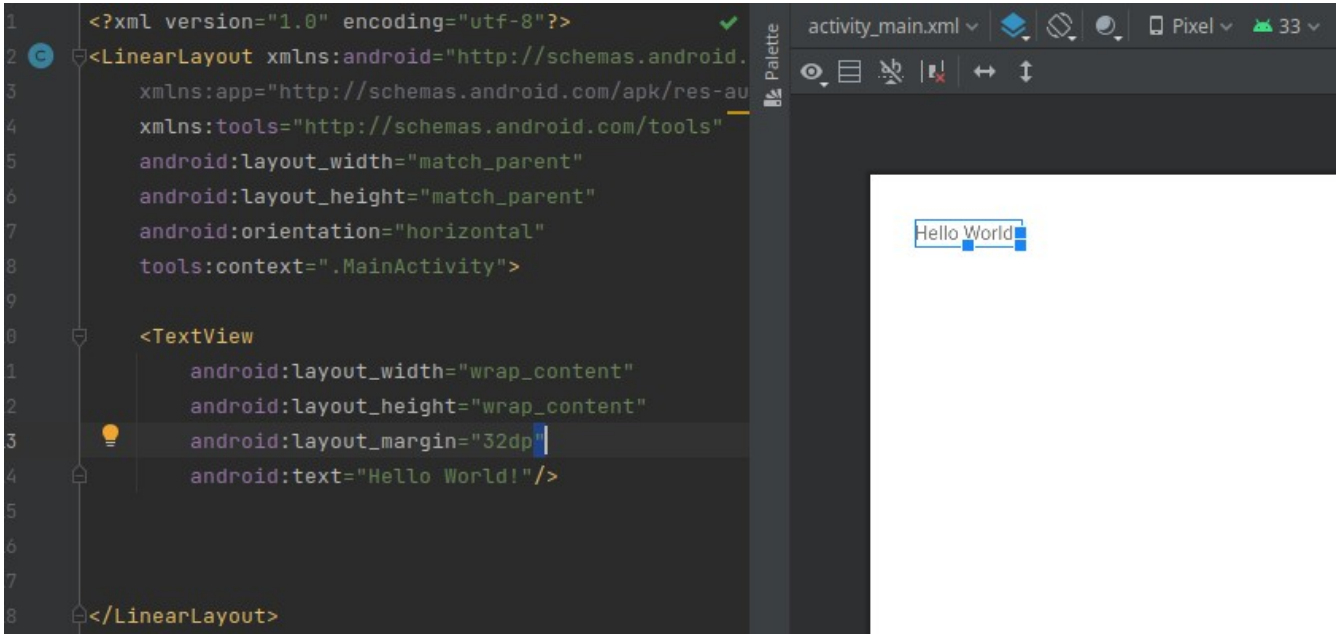
Atributos de Layout

- layout_margin
  - Usar sempre múltiplos de 2 para especificar as margens: 8dp, 16dp, 32dp, 64dp e 128dp



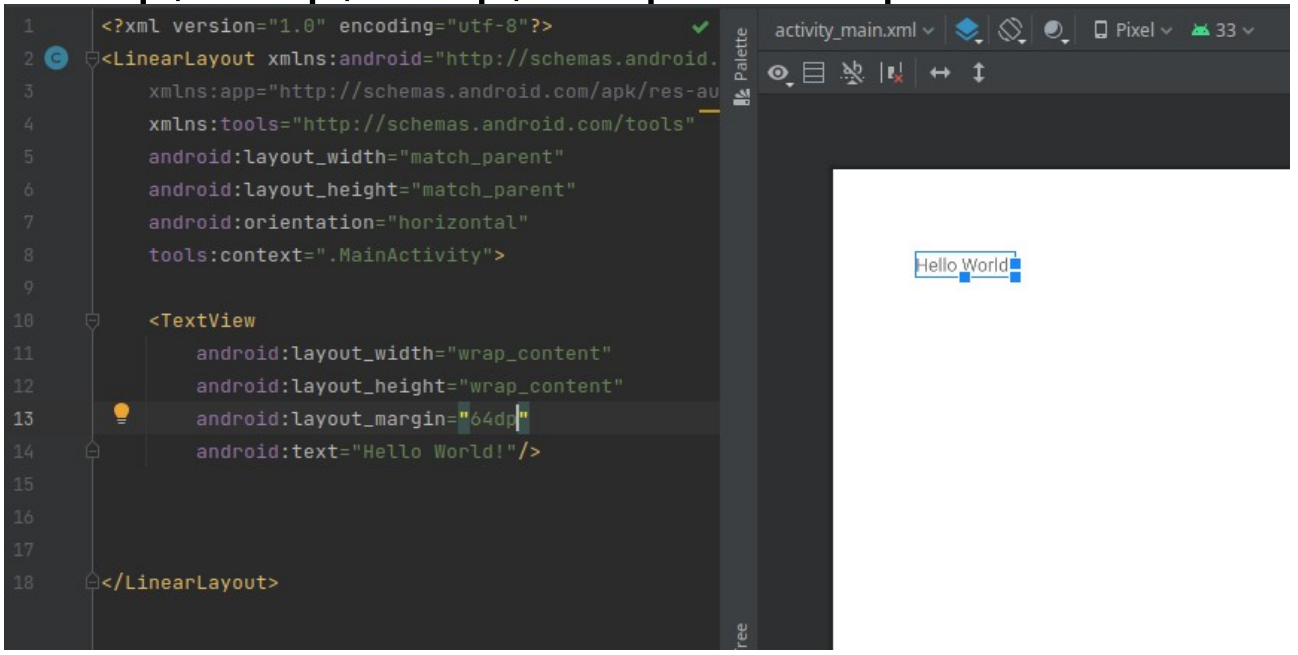
Atributos de Layout

- layout_margin
  - Usar sempre múltiplos de 2 para especificar as margens: 8dp, 16dp, 32dp, 64dp e 128dp



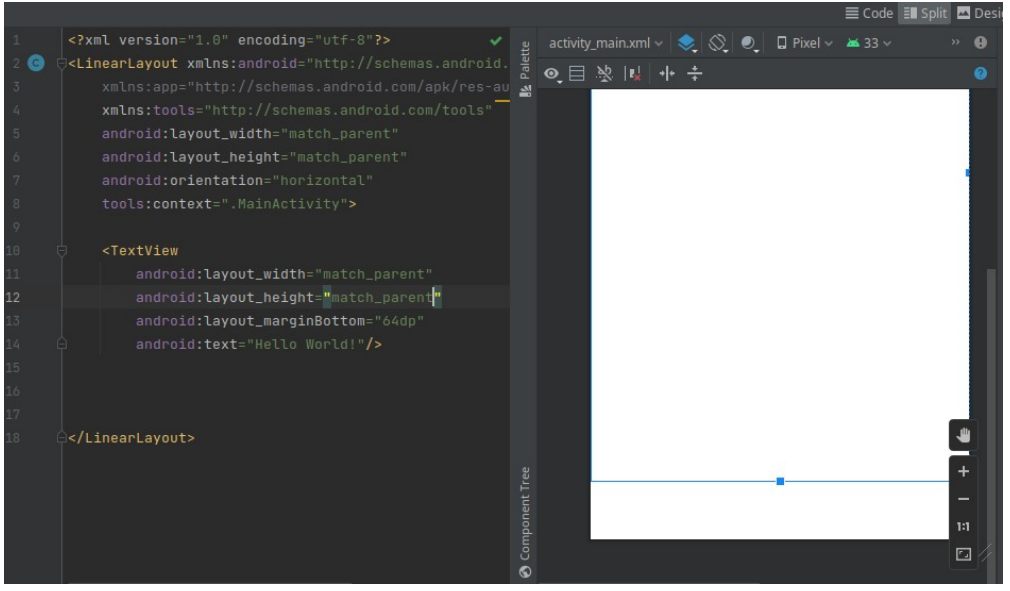
Atributos de Layout

- layout_margin
  - Usar sempre múltiplos de 2 para especificar as margens: 8dp, 16dp, 32dp, 64dp e 128dp



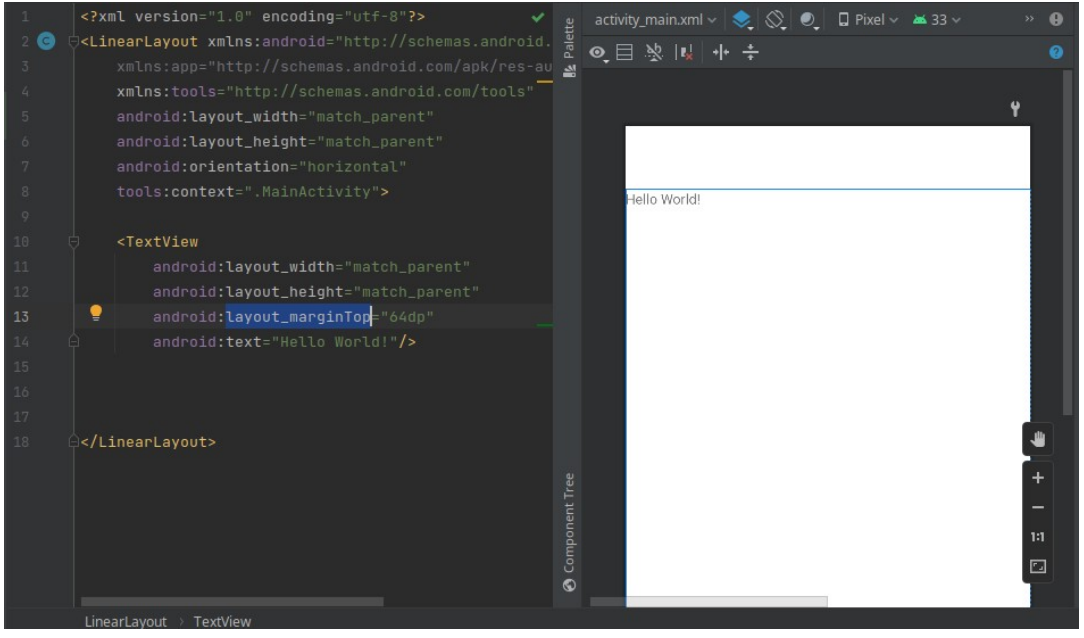
Atributos de Layout

- layout_marginBottom
  - Atribui um espaçamento entre o laytout e o elemento na parte inferior do layout



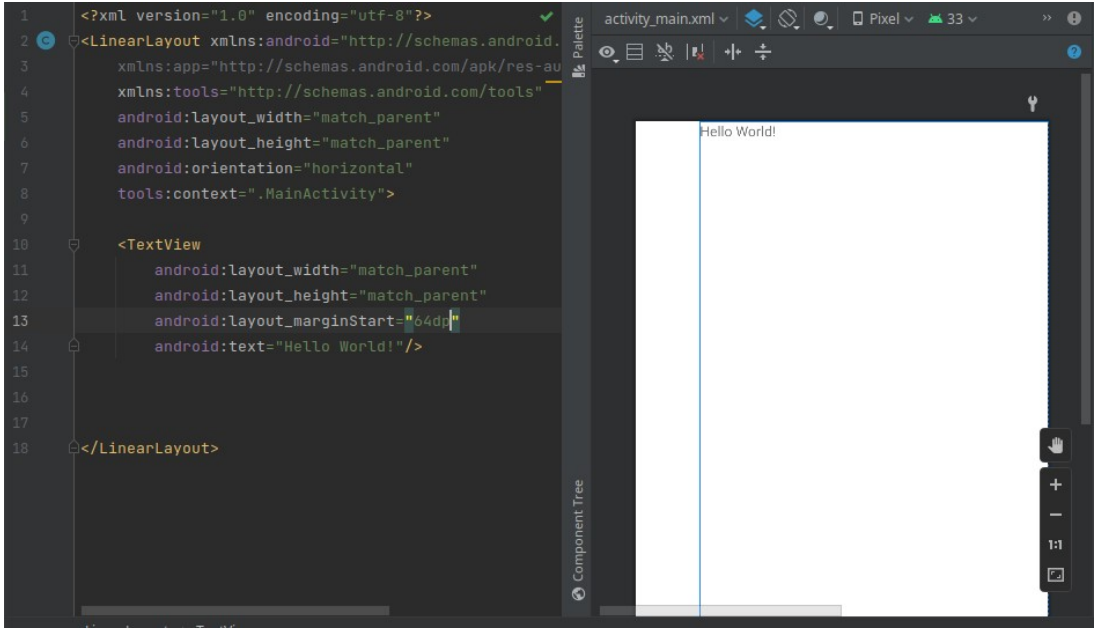
Atributos de Layout

- layout_marginTop
  - Atribui um espaçamento entre o laytout e o elemento na parte superior do layout



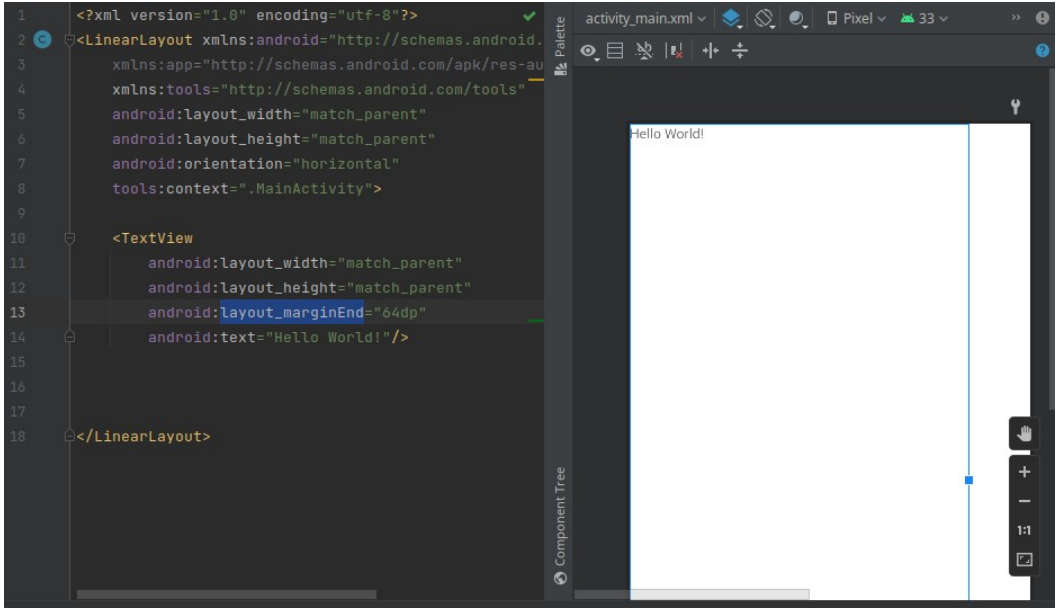
Atributos de Layout

- layout_marginStart
  - Atribui um espaçamento entre o laytout e o elemento do lado esquerdo do layout



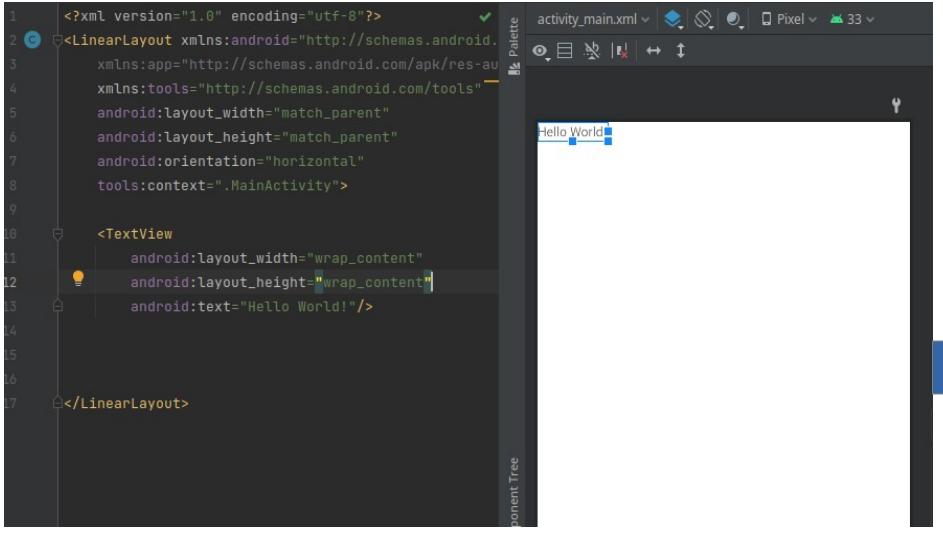
Atributos de Layout

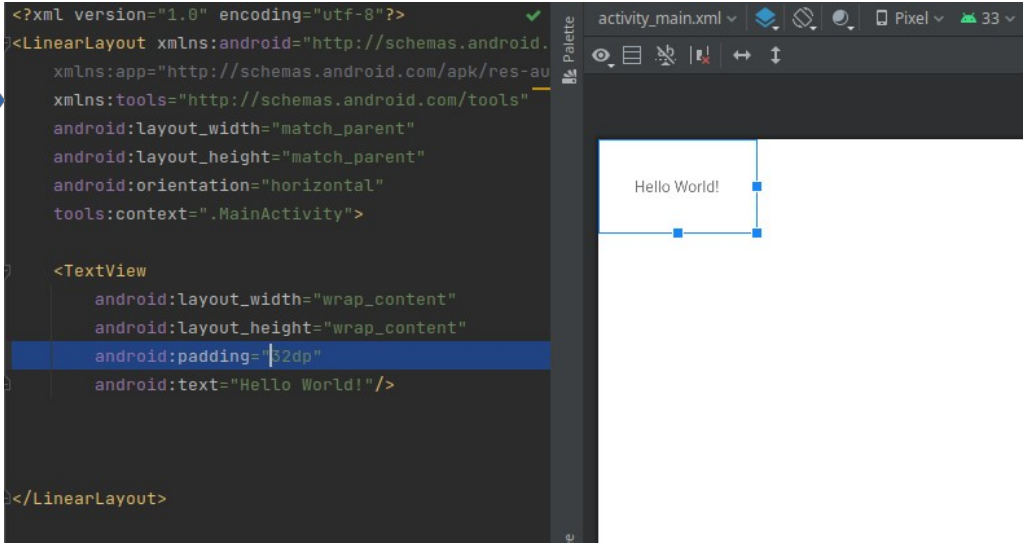
- layout_marginEnd
  - Atribui um espaçamento entre o laytout e o elemento do lado direito do layout



Atributos de Layout

Aumentando o tamanho dos elementos com **padding:**



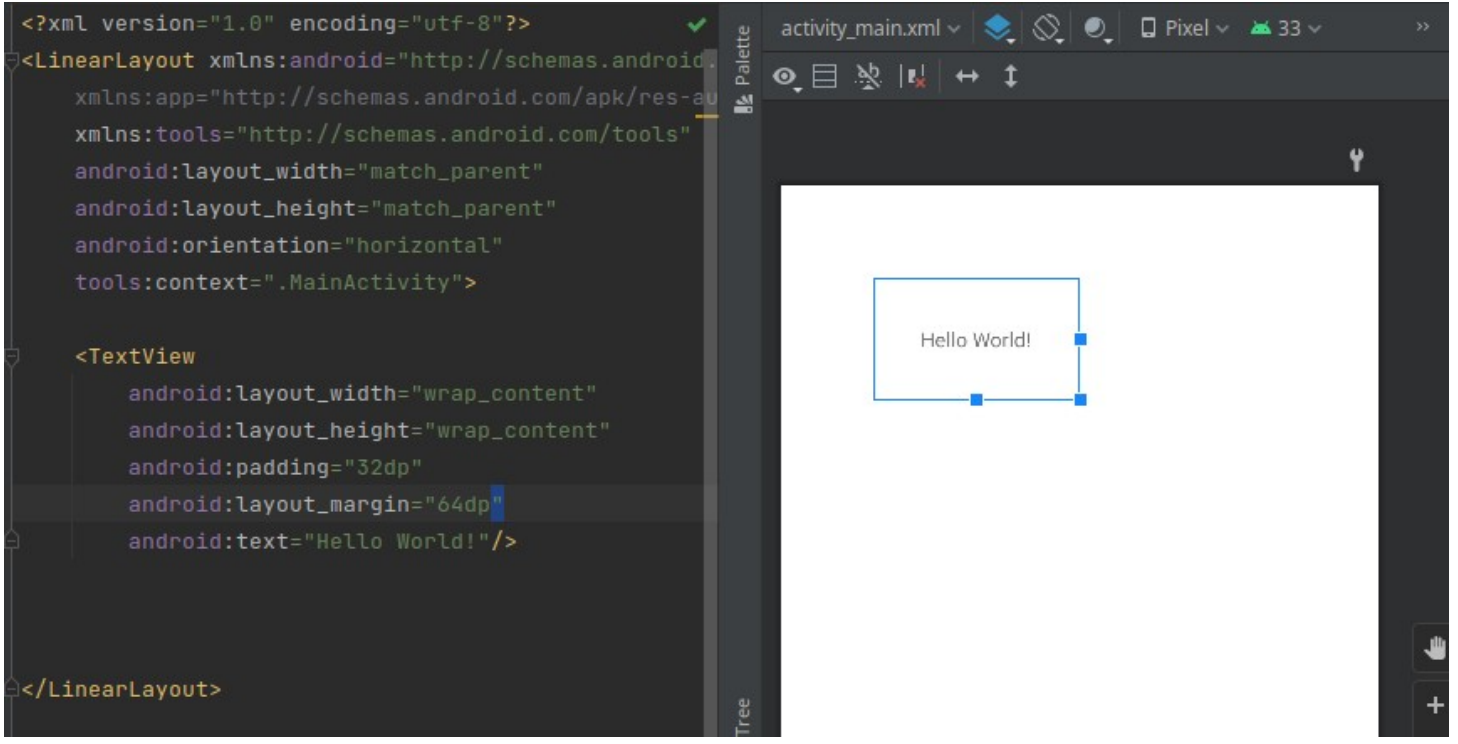


Atributos de Layout

Os atributos de layout **layout_marginStart** e **layout_margiEnd** foram introduzidos nas versões mais recentes do android e substituem os atributos **layout_marginLeft** e **layout_marginRight**

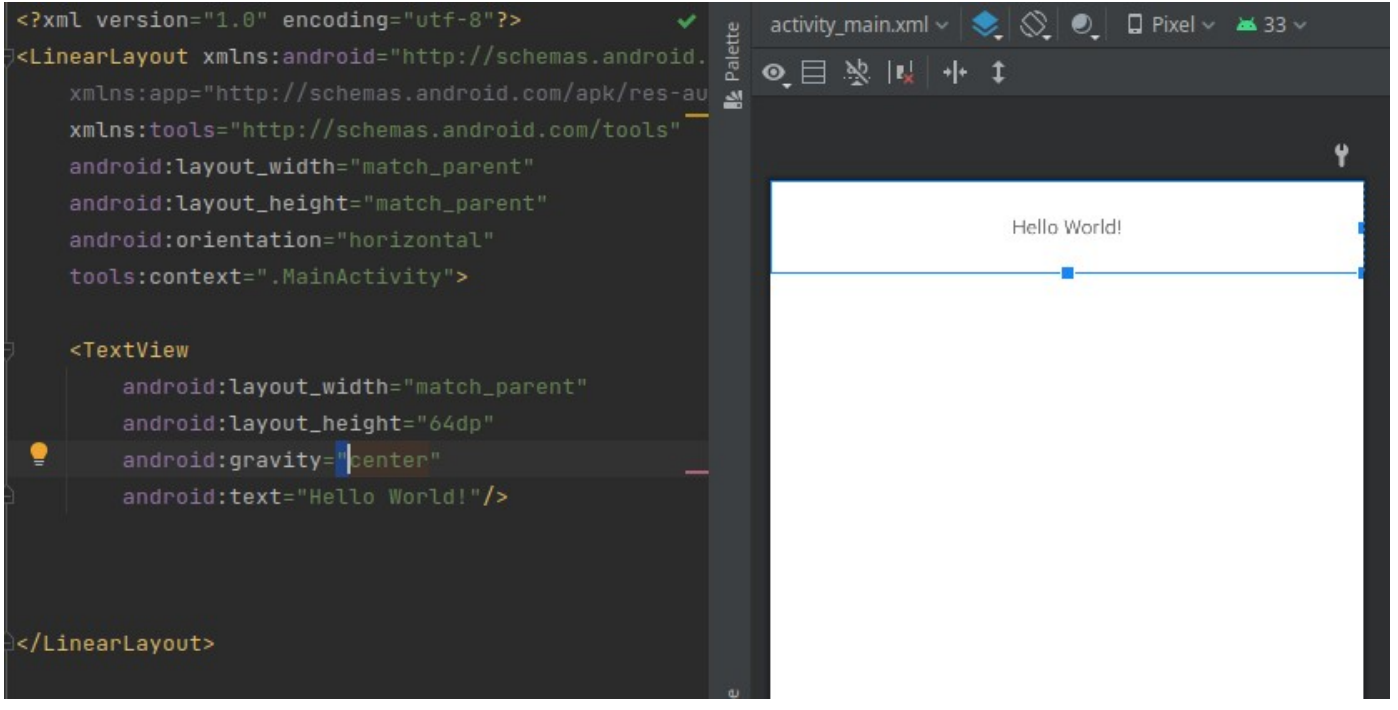
Atributos de Layout

Usando **padding** e **layout_margin:**



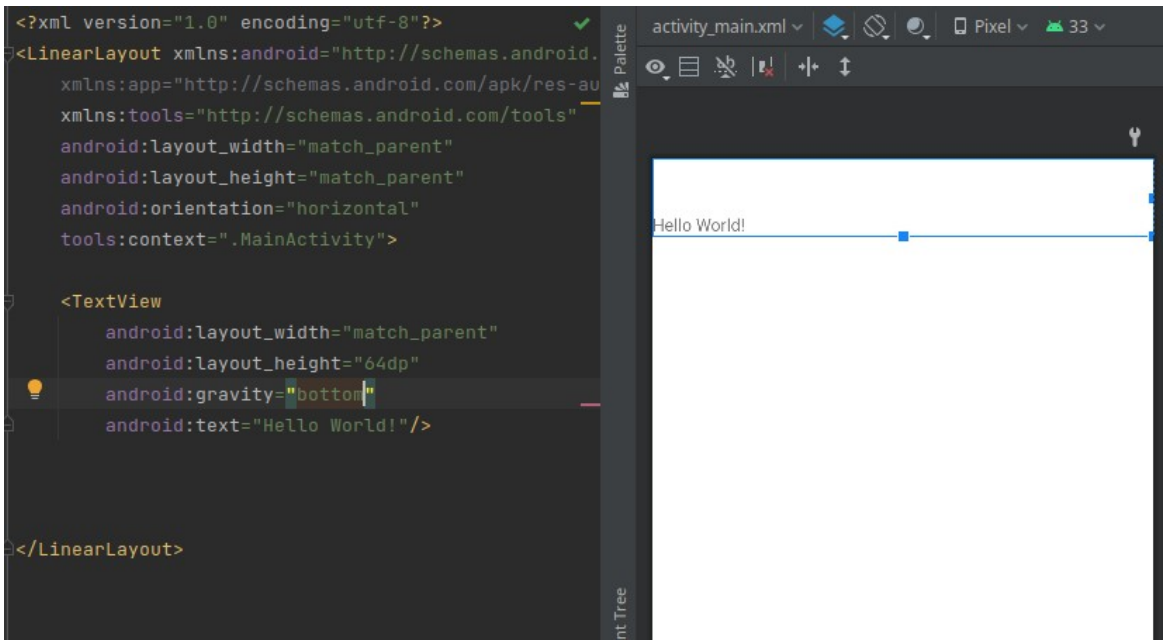
Gravity

- Alinhamento dentro do elemento:
  - **android:gravity=”center”**



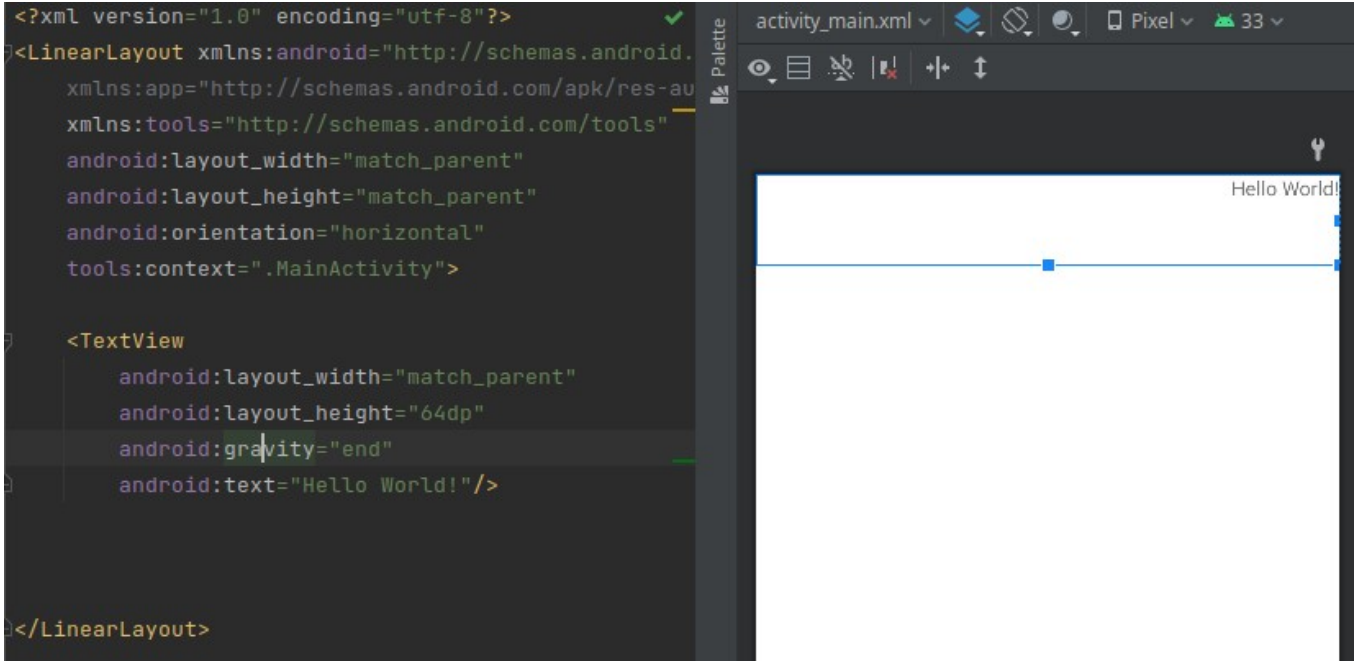
Gravity

- Alinhamento dentro do element:
  - **android:gravity=”bottom”**



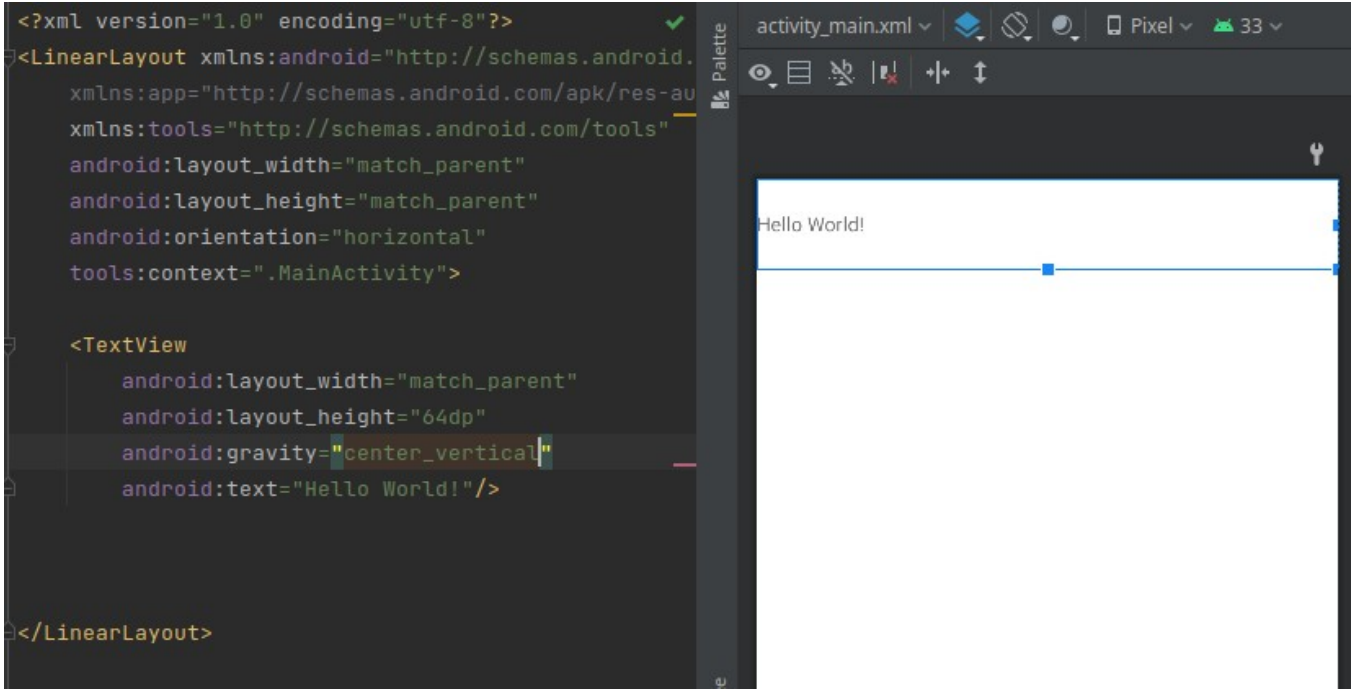
Gravity

- Alinhamento dentro do element:
  - **android:gravity=”end”**



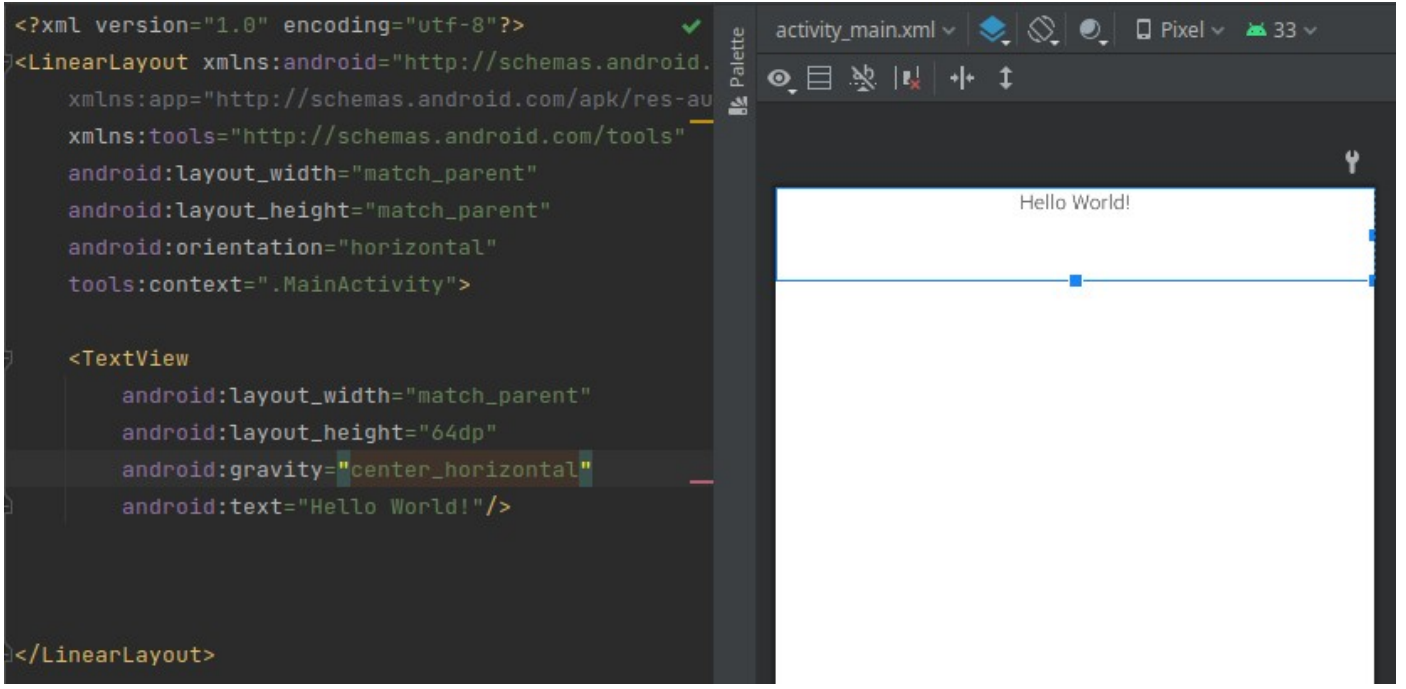
Gravity

- Alinhamento dentro do element:
  - **android:gravity=”center-vertical”**



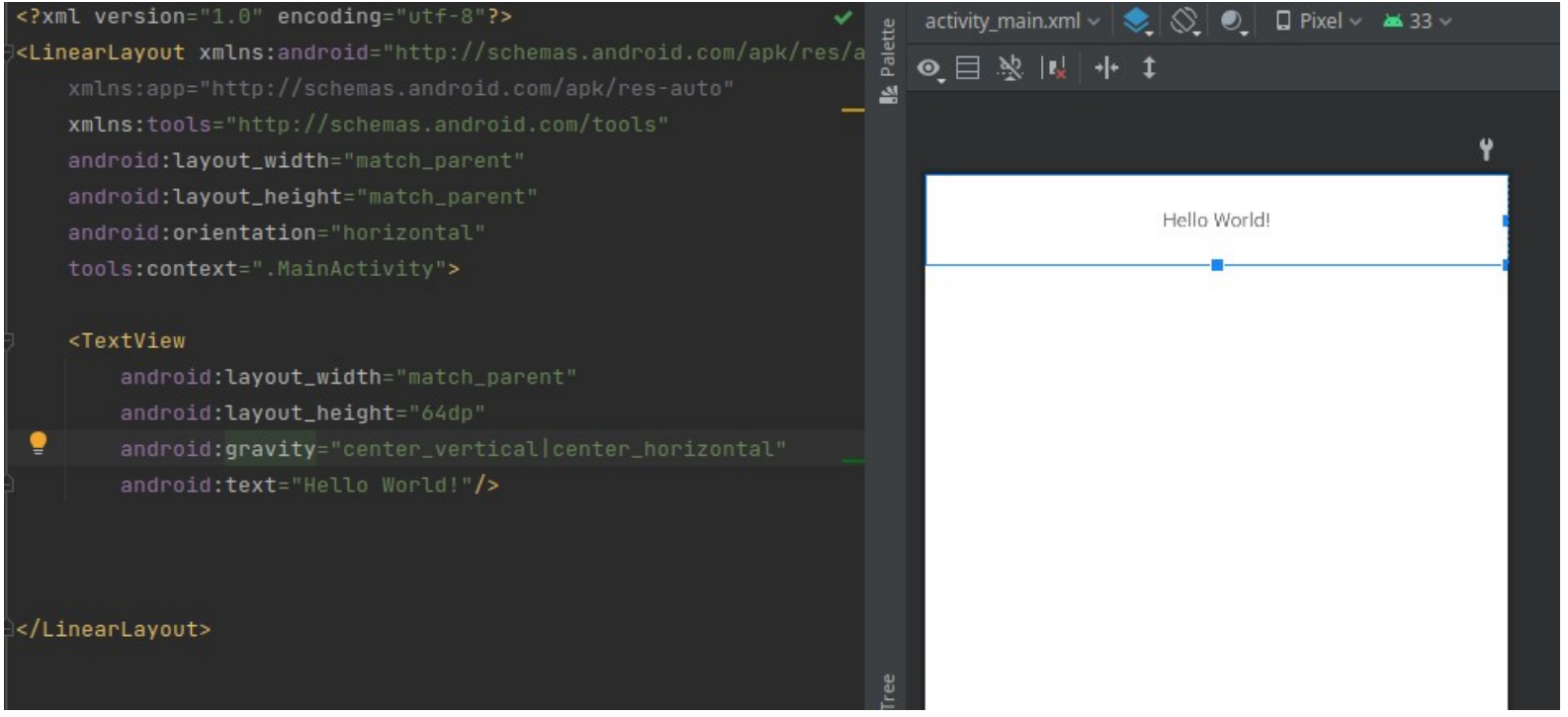
Gravity

- Alinhamento dentro do element:
  - **android:gravity=”center-horizontal”**



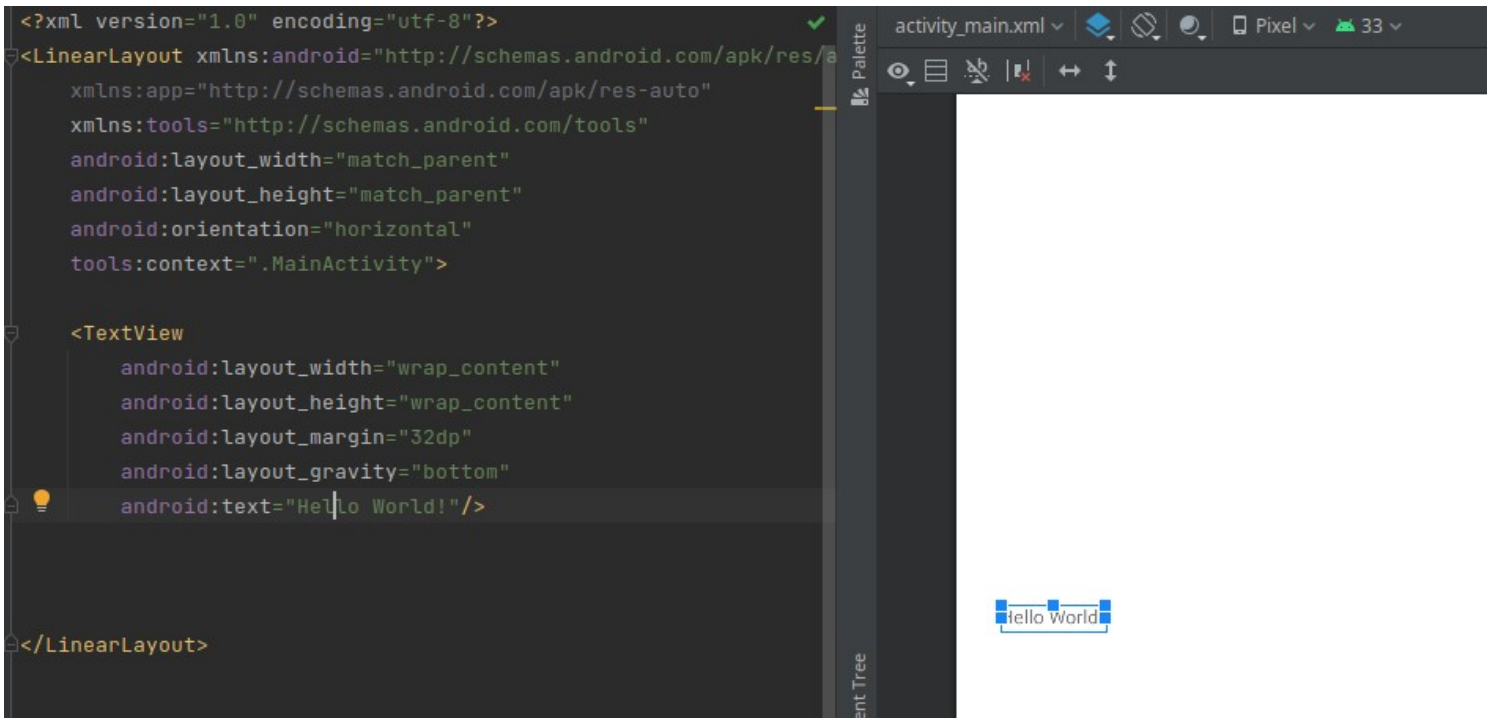
Gravity

- Alinhamento dentro do element:
  - **android:gravity=”center_vertical|center_horizontal”**



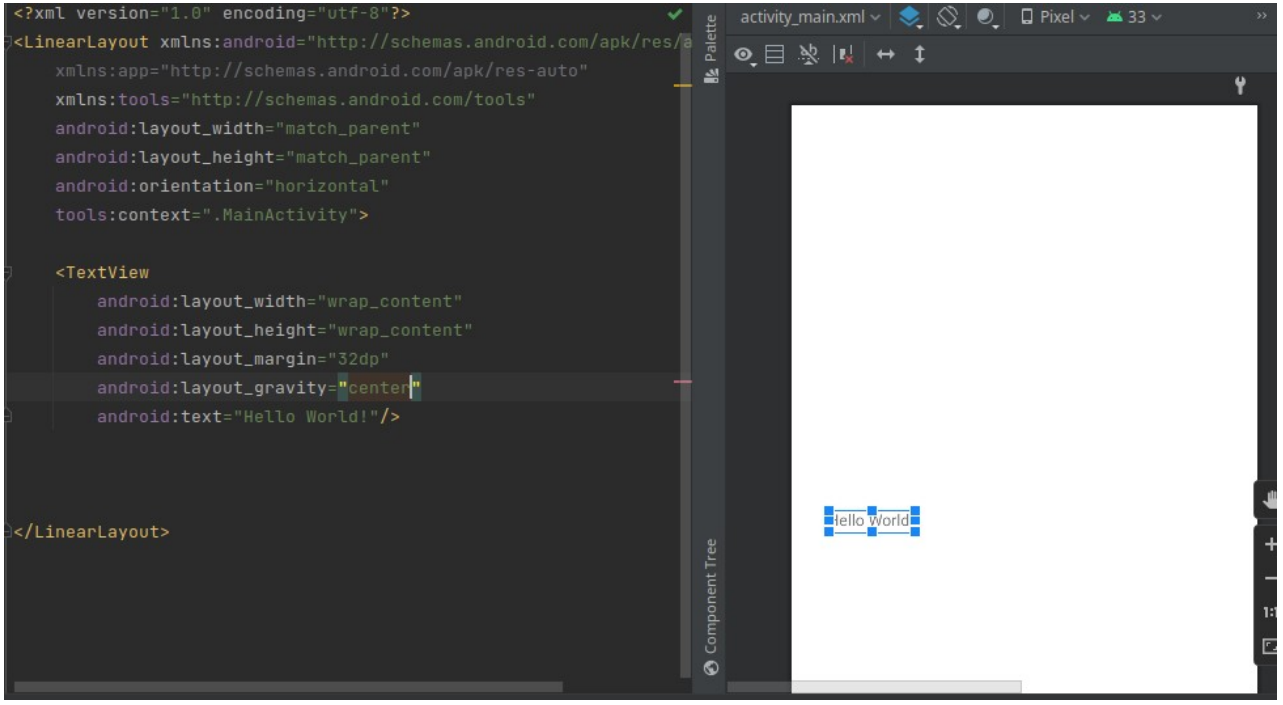
layout_gravity

- Alinhamento do elemento em si em relação ao elemento pai.



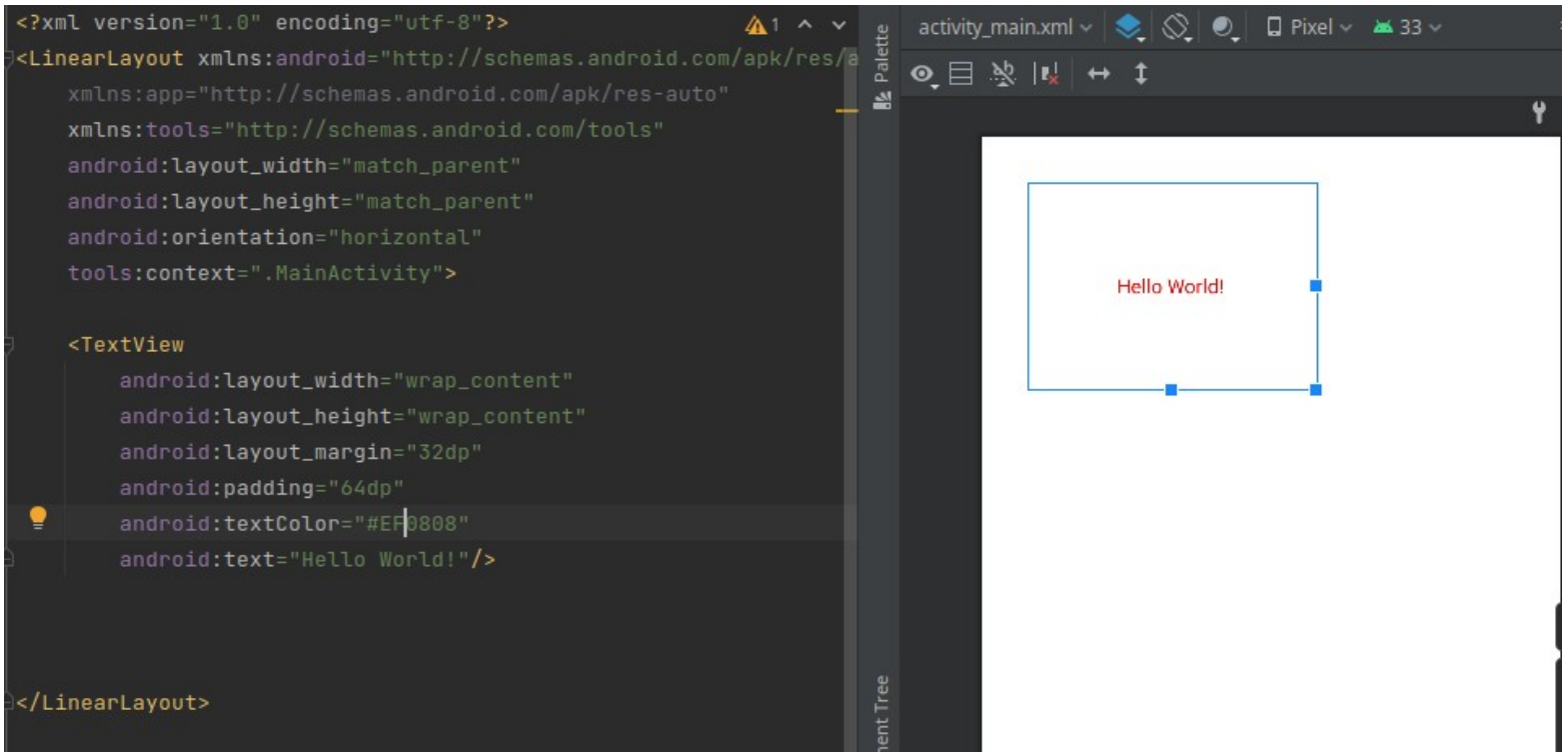
layout_gravity

- Alinhamento do elemento em si em relação ao elemento pai.



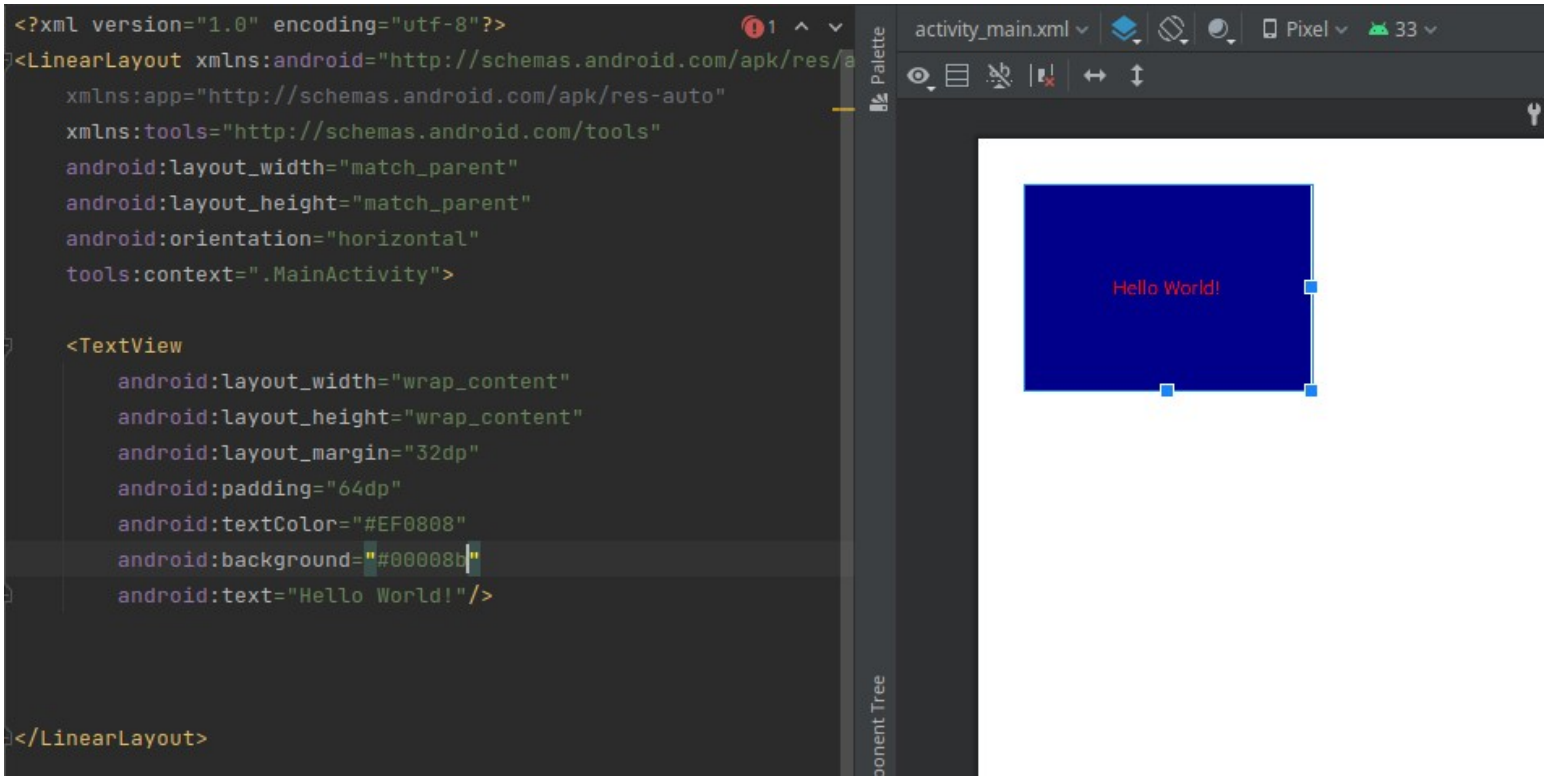
textColor

Altera a cor da fonte do elemento.



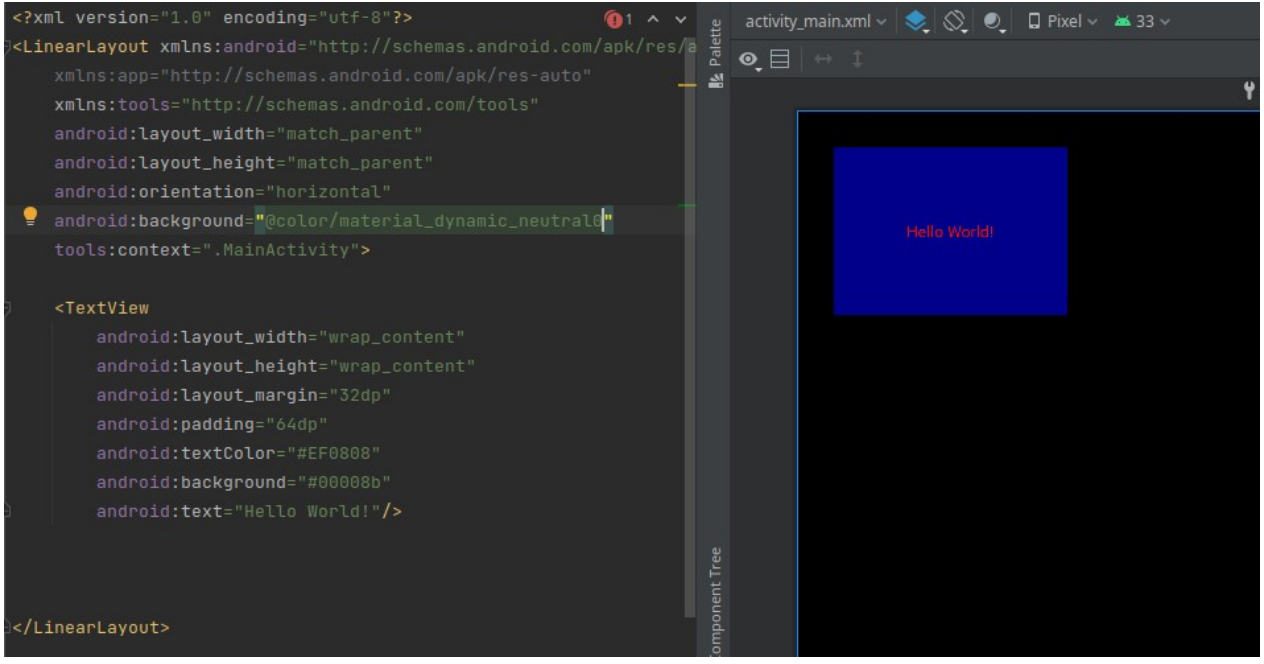
background

- Altera a cor do plano de fundo do elemento.

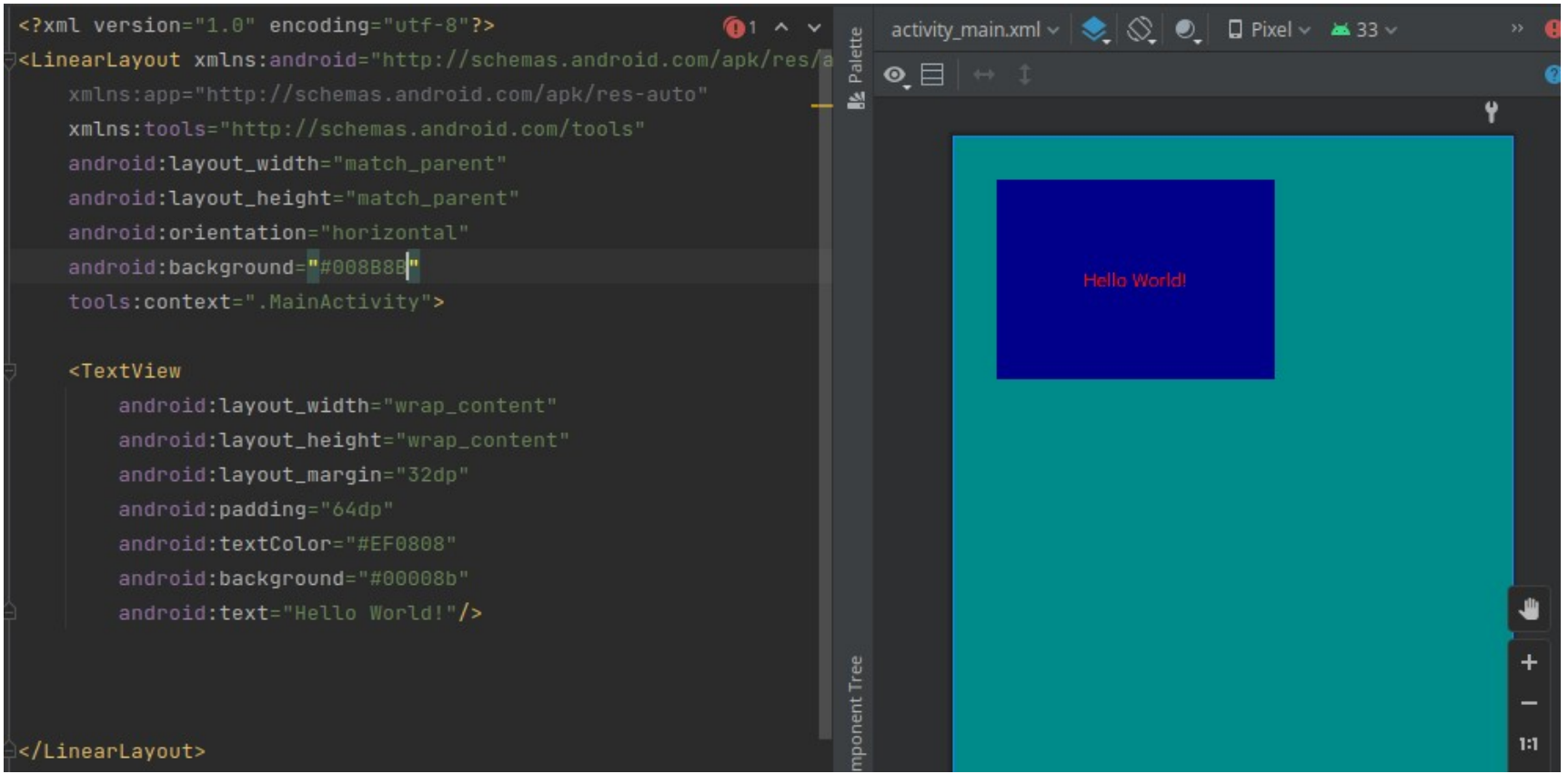


background

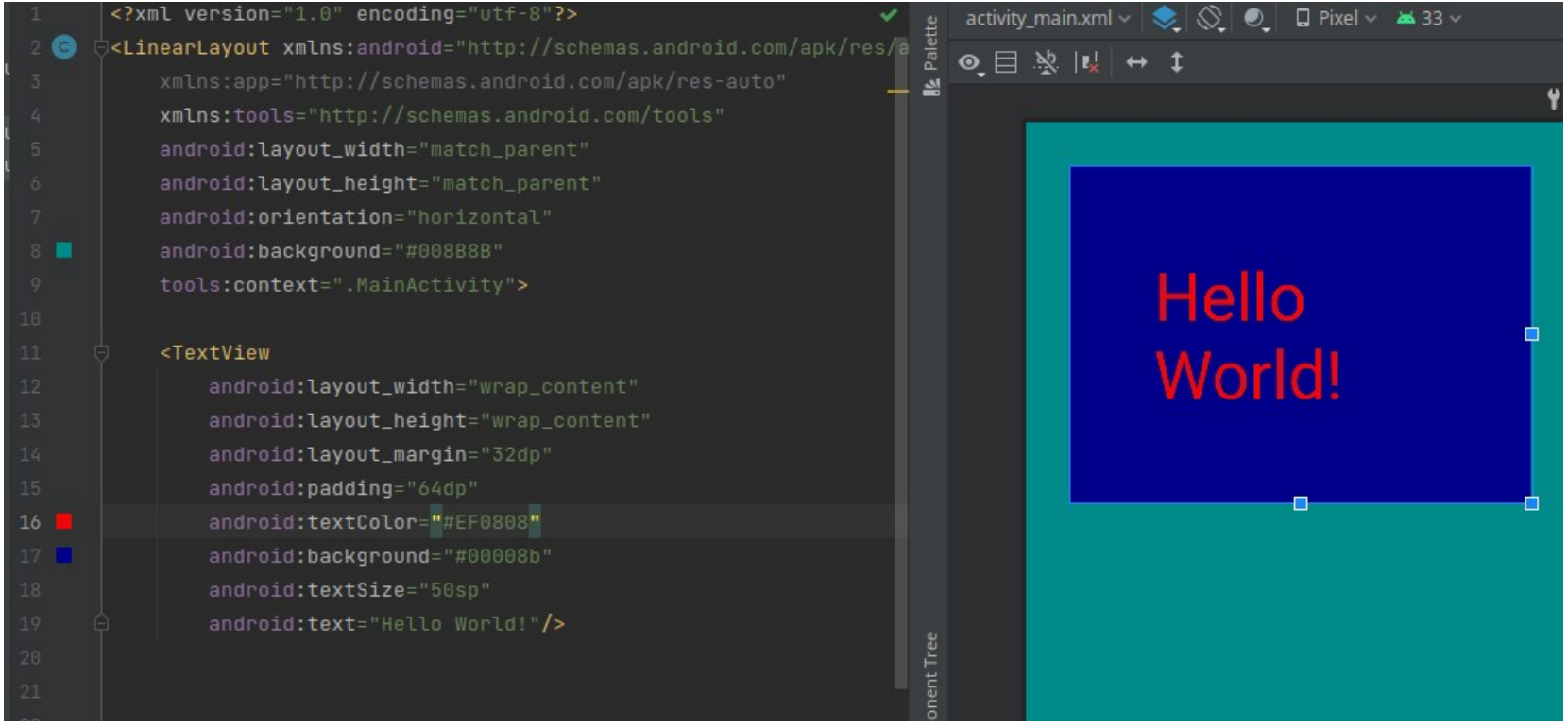
- Como o elemento Pai **LinearLayout** não tem texto não é possível alterar cor de texto, mas podemos alterar a cor do fundo com **background**.



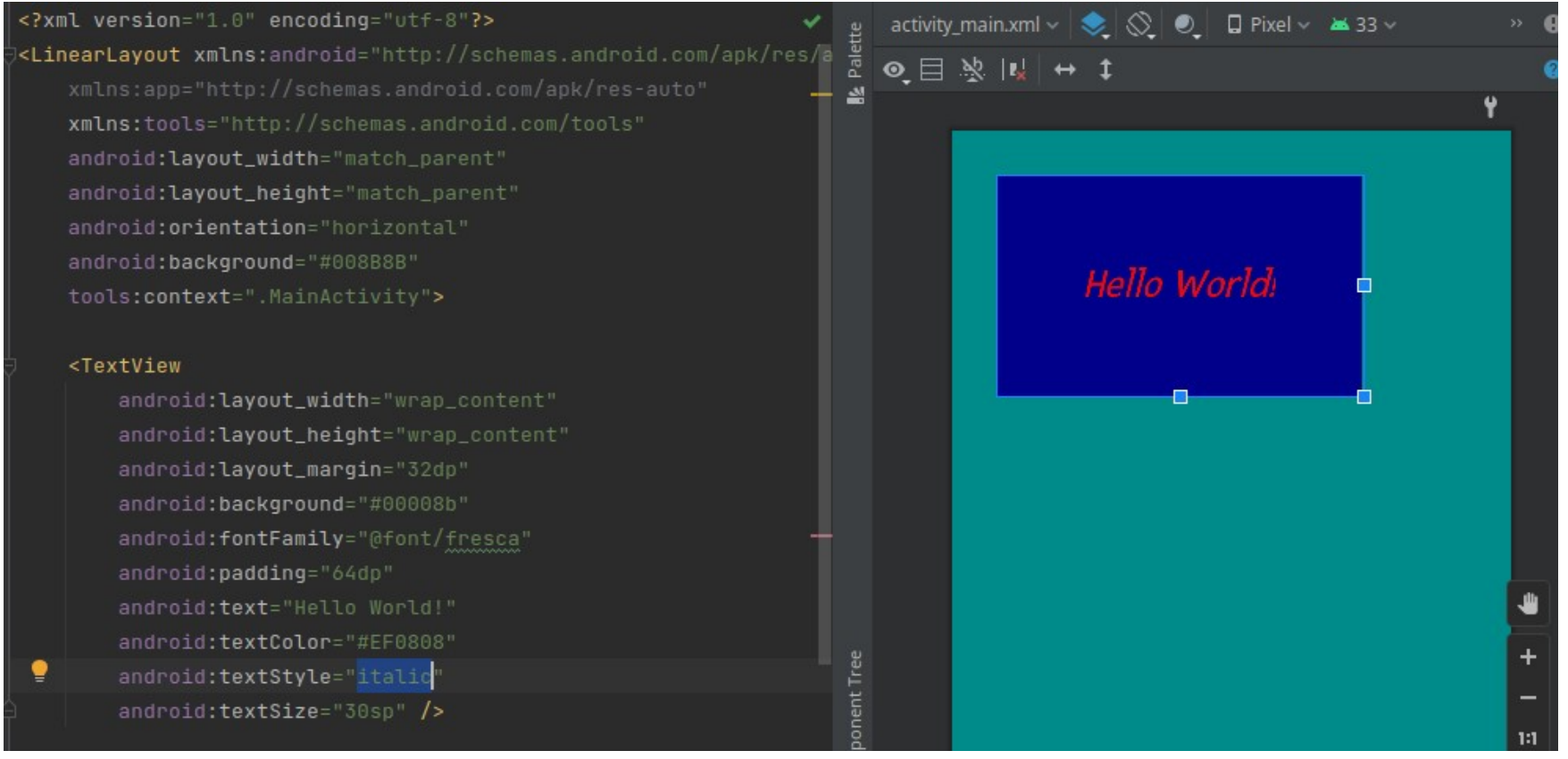
background


<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/a
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="horizontal"
android:background="#008B8B"
tools:context=".MainActivity">
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="32dp"
android:padding="64dp"
android:textColor="#EF0808"
android:background="#00008b"
android:text="Hello World!"/>
</LinearLayout>
Palette
activity_main.xml
Pixel
33
Hello World!
1:1

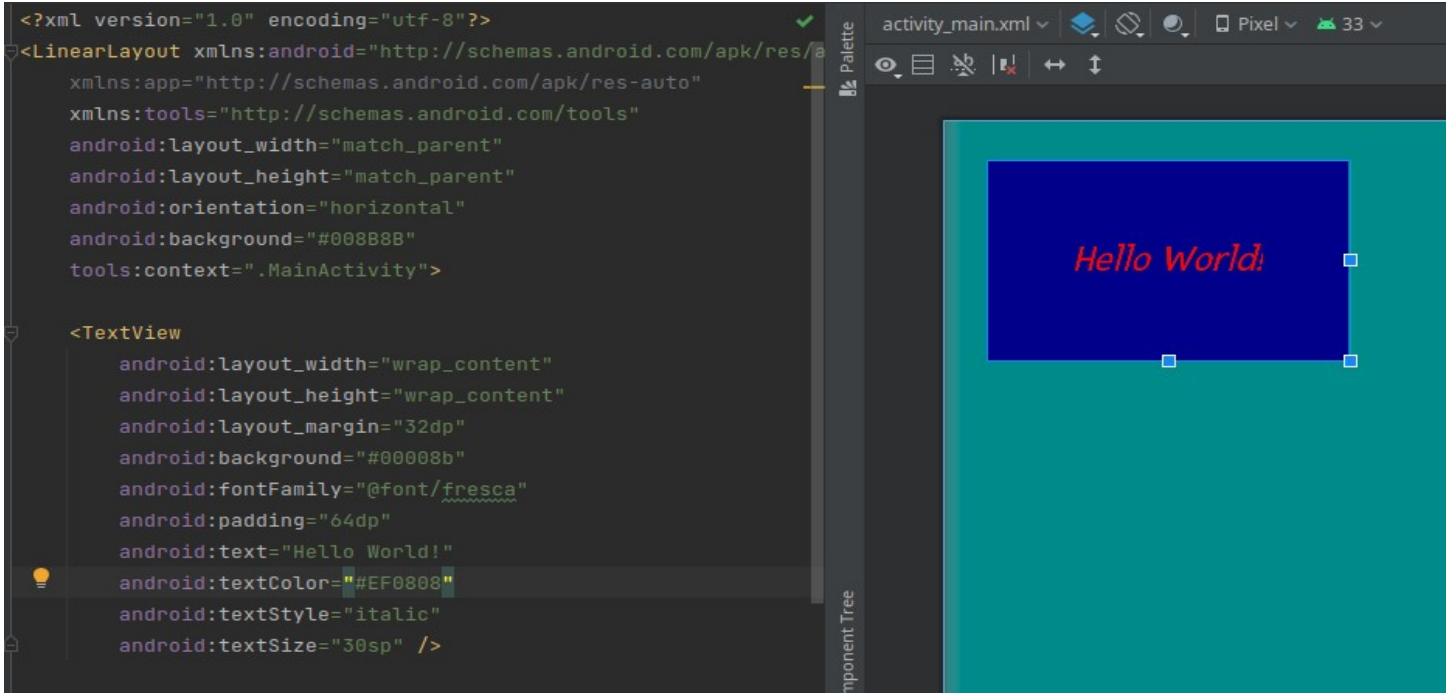
Aumentando Tamanho da Fonte



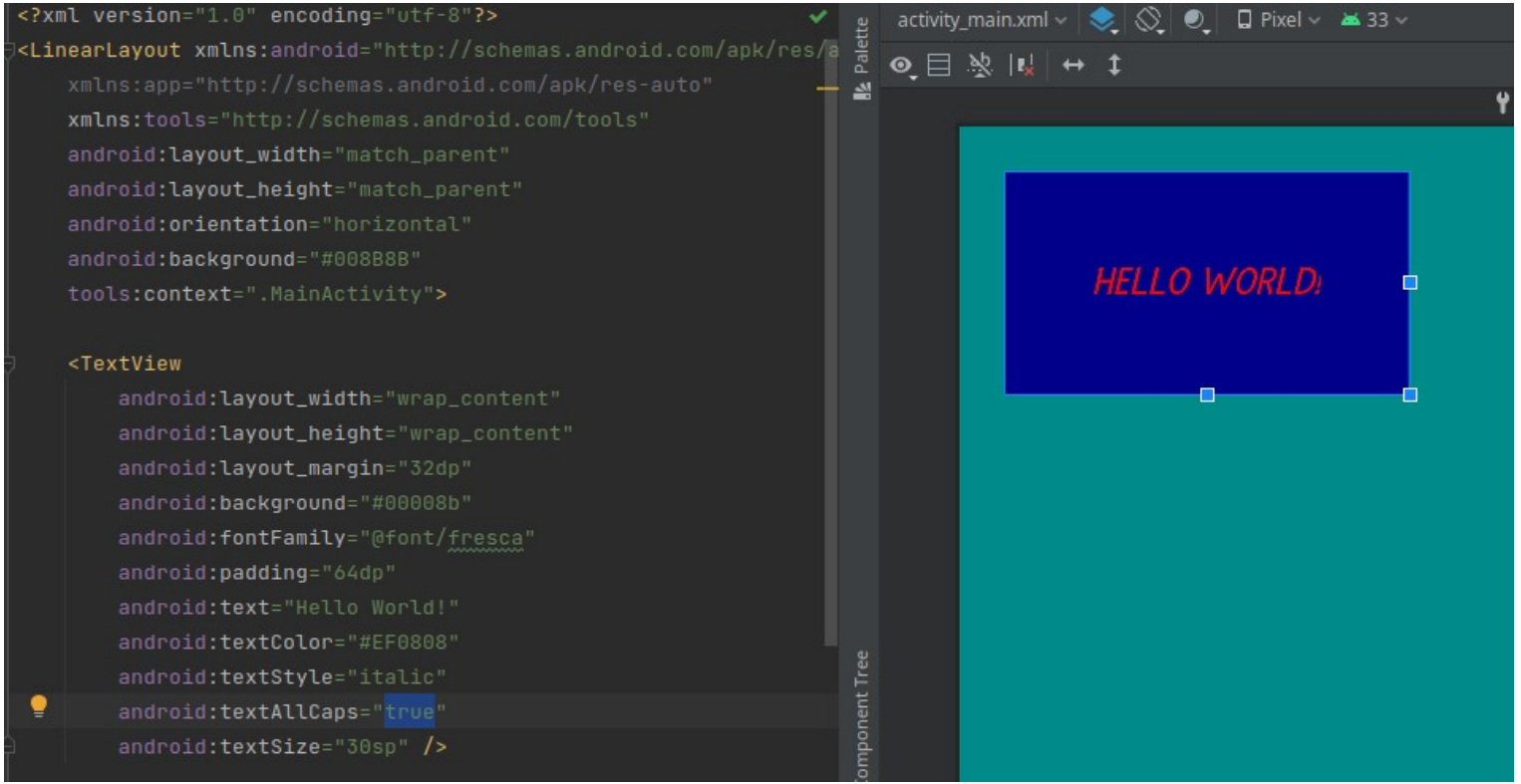
Alterando o tipo da fonte



Alterando o estilo da fonte



Colocando a fonte em maiúsculo



Referências

LECHETA, Ricardo. R. Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK. 5ª ed. São Paulo, SP: Novatec, 2016.