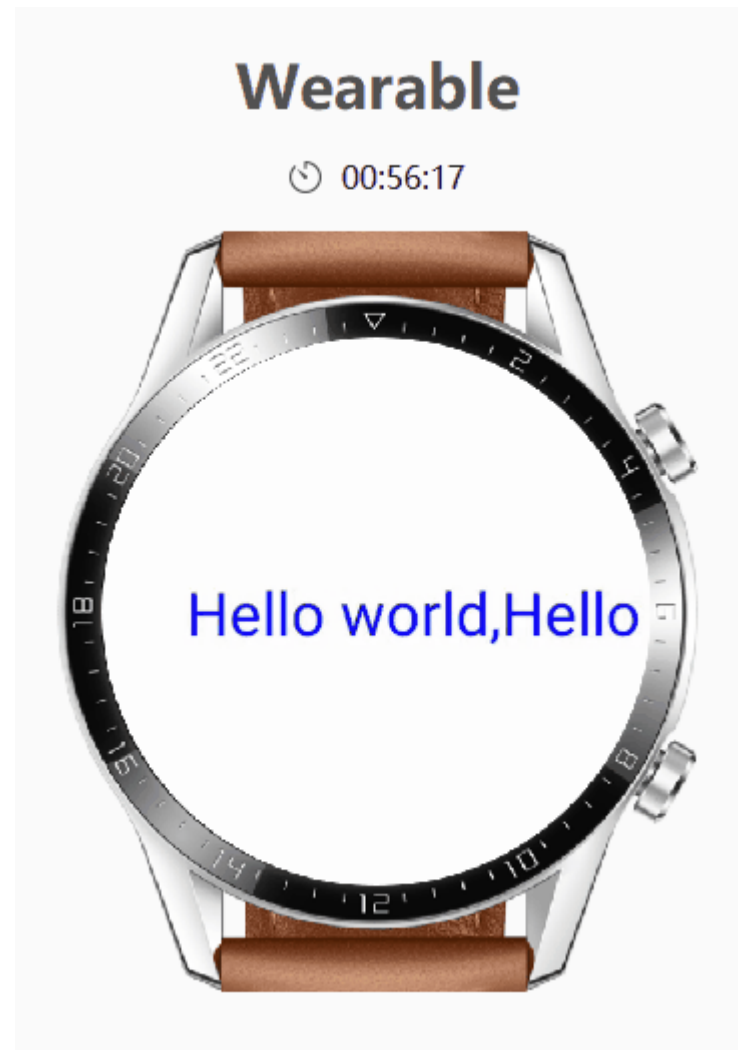


Harmony OS 入门系列课程 <快速上手>

快速掌握鸿蒙系统应用开发基础操作技巧

第 2 讲：UI组件Text快速实现跑马灯



本讲内容

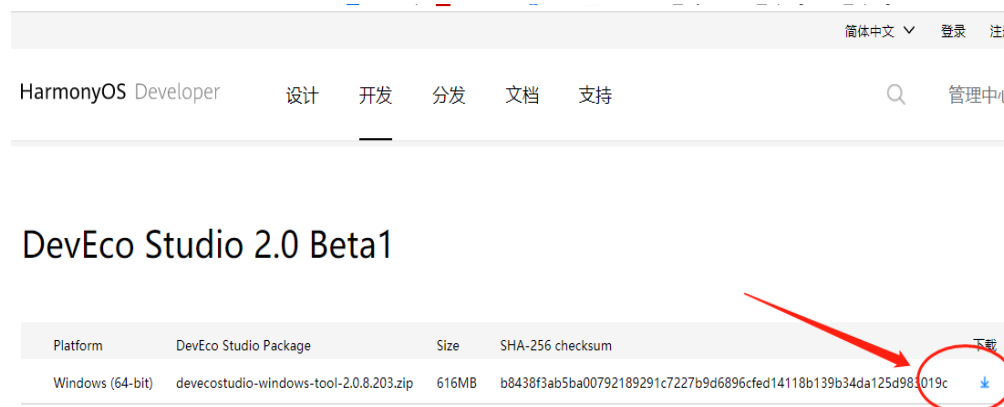
- 知识点1: java代码快速实现跑马灯 (重点)
- 知识点2: 详细分析跑马灯代码 (难点)
- 知识点3: 代码创建布局拓展 (难点)
- 知识点4: 通过XML创建布局实现跑马灯 (重点)

本讲目标

- 能够快速实现跑马灯功能
- 理解跑马灯中每行代码的作用
- 能够用java代码创建布局;
- 能够用XML文件创建布局;

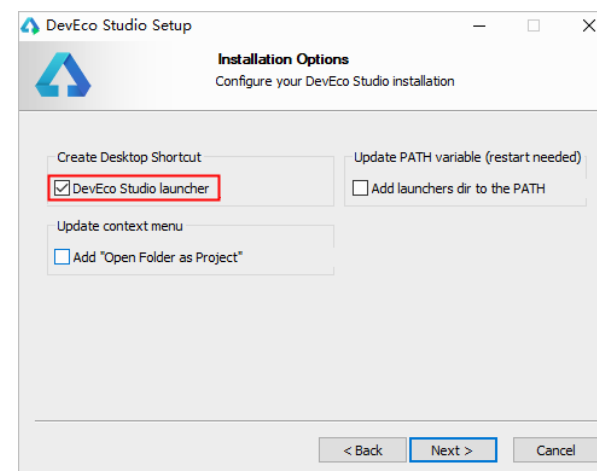
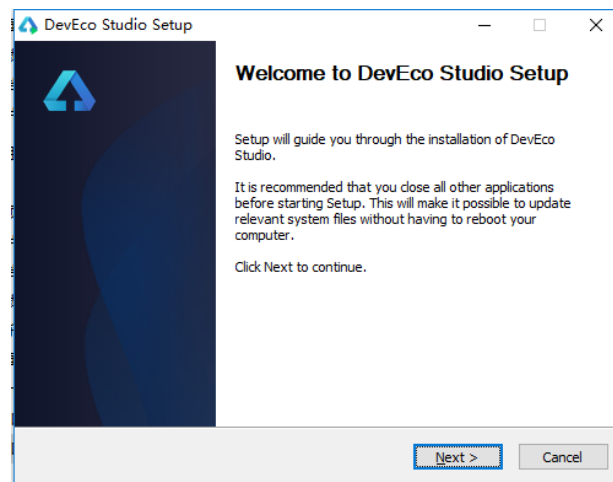
上讲回顾【下载和安装DevEco Studio】

- 官方下载地址: <https://developer.harmonyos.com/cn/develop/deveco-studio>



■ 安装DevEco Studio

名称	修改日期	类型	大小
deveco-studio-2.0.8.203.exe	2020/9/6 13:38	应用程序	631,520 KB



上讲回顾【创建及运行Hello World工程】

操作演示



知识点1 【java代码快速实现跑马灯】

在已经创建好的Hello World工程上添加完成跑马灯代码

<code/>

```
//设置当前文本内容超出显示屏大小时，文本截断方式
text.setTruncationMode(Text.TruncationMode.AUTO_SCROLLING);
//开启文本内容滚动
text.startAutoScrolling();
//设置滚动次数，-1表示永远不停
text.setAutoScrollingCount(-1);
```

知识点1 【java代码快速实现跑马灯】

操作演示



知识点2 【详细分析跑马灯代码】

布局代码分析

<code/>

```
DirectionalLayout myLayout = new DirectionalLayout(this); //创建线性布局组件
//创建布局配置对象， MATCH_PARENT子组件大小填充适配父布局组件
LayoutConfig config = new LayoutConfig(LayoutConfig.MATCH_PARENT, LayoutConfig.MATCH_PARENT);
myLayout.setLayoutConfig(config); //给布局组件设置大小

ShapeElement element = new ShapeElement(); //创建形状类型元素
element.setRgbColor(new RgbColor(255, 255, 255)); //设置颜色
myLayout.setBackground(element); //设置布局组件背景色
```


知识点2 【详细分析跑马灯代码】

组件代码分析

<code/>

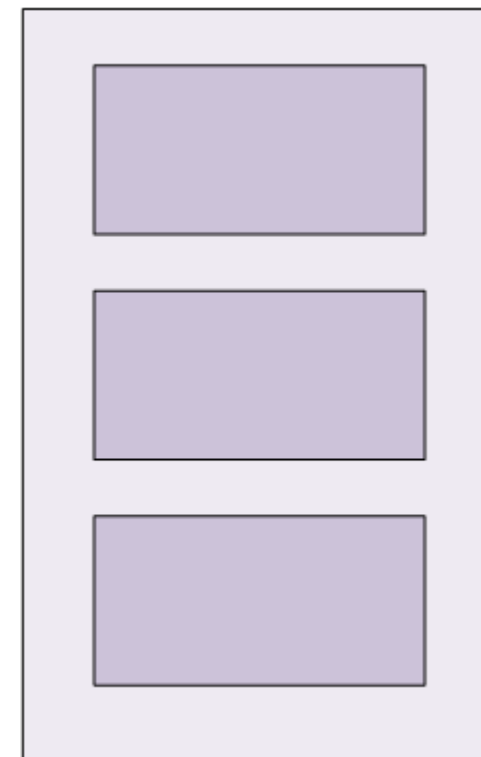
```
Text text = new Text(this); //创建Text组件
text.setLayoutConfig(config); //设置text的大小
text.setText("Hello world,Hello 鸿蒙，我们来了，我们陪伴你成长！"); //设置text显示内容
text.setTextColor(new Color(0xFF0000ff)); //设置字体颜色
text.setTextSize(50); //设置字体大小
text.setTextAlignment(TextAlignment.CENTER); //设置文本内容对齐方式

myLayout.addComponent(text); //把text组件添加到布局组件中
```

疑问

此时你应该有许多疑问：

- 疑问1：一个组件可以填充父控件，那有二个、三个，甚至更多时如何实现？
- 疑问2：如何控制布局呢？
- DirectionalLayout如何实现？



知识点3 【代码创建布局拓展】

DirectionalLayout, 是Java UI中的一种重要组件布局, 用于将一组组件(Component)按照水平或者垂直方向排布, 能够方便地对齐布局内的组件, DirectionalLayout的排列方向 (orientation [ˌɔːriənˈteɪʃn]) 分为:

- horizontal [ˌhɒːrɪˈzɔːntl] : 水平方向排列
- vertical [ˈvɜːrtɪkl] : 垂直方向排列

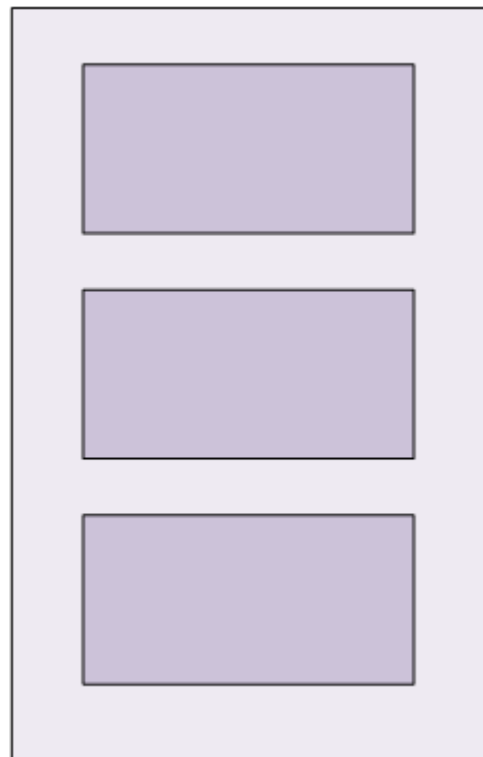
知识点3 【代码创建布局拓展】

实现垂直方向排列三个Text文本组件

<code/>

```
DirectionalLayout myLayout = new DirectionalLayout(this); //创建线性布局组件
//设置线性布局组件DirectionalLayout的排列方向为垂直
myLayout .setOrientation(Component.VERTICAL);

//添加布局组件中添加3个Text组件
myLayout.addComponent(text);
myLayout.addComponent(text2);
myLayout.addComponent(text3);
```



知识点3 【代码创建布局拓展】

操作演示



疑问

此时你应该有许多疑问：

- 疑问1：所有的代码都放到一个文件中？
- 疑问2：UI布局和逻辑实现能分开吗？
- 疑问3：UI布局如何实现

知识点4 【通过XML创建布局实现跑马灯】

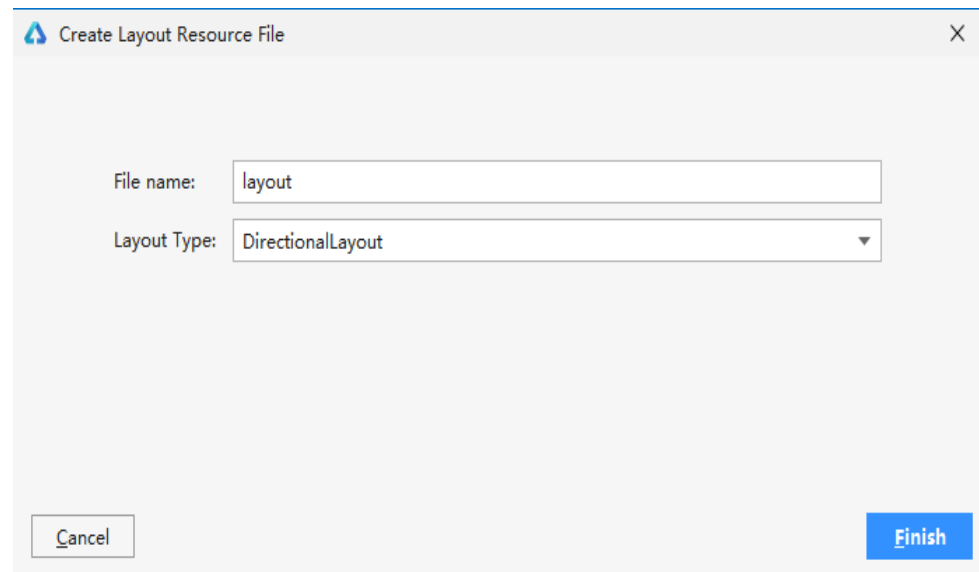
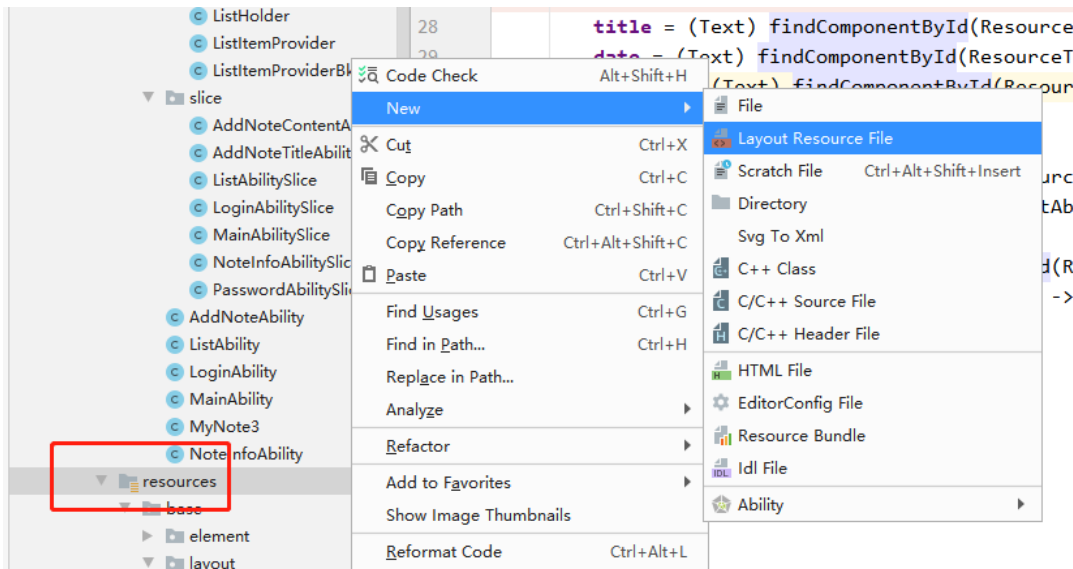
实现步骤：

1. 创建XML布局文件；
2. 修改XML布局文件；
3. 代码中完成加载XML布局并开启滚动功能

知识点4【通过XML创建布局实现跑马灯】

步骤1：创建XML布局文件

1. 在DevEco Studio的“Project”窗口，打开“entry > src > main > ”，右键点击“resources”文件夹，选择“New > Layout Resource File”，命名为“layout”



知识点4 【通过XML创建布局实现跑马灯】

步骤2：修改XML布局文件；

<code/>

```
<?xml version="1.0" encoding="utf-8"?>
<DirectionalLayout xmlns:ohos="http://schemas.huawei.com/res/ohos"
    ohos:width="match_parent"
    ohos:height="match_parent"
    ohos:background_element="#FFFFFFFF"
    ohos:orientation="vertical">
```

//此处为Text组件，见下一页PPT

```
</DirectionalLayout>
```

知识点4 【通过XML创建布局实现跑马灯】

步骤2：修改XML布局文件；

<code/>

```
<Text
    ohos:id="$+id:text1"
    ohos:width="match_parent"
    ohos:height="match_parent"
    ohos:text_alignment="vertical_center"
    ohos:text="鸿蒙， 我们来啦！ 中软卓越陪伴你成长！ "
    ohos:text_color="blue"
    ohos:text_size="50fp"
    ohos:multiple_lines="false"
    ohos:max_text_lines="1" />
```

知识点4【通过XML创建布局实现跑马灯】

步骤3：代码中完成加载XML布局并开启滚动功能

<code/>

```
setUIContent(ResourceTable.Layout_layout);  
Text text = (Text) findComponentById(ResourceTable.Id_text1);  
text.setTruncationMode(Text.TruncationMode.AUTO_SCROLLING);  
text.startAutoScrolling();  
text.setAutoScrollingCount(-1);
```



本讲小结

本讲所学知识点有：

- 知识点1：java代码快速实现跑马灯
- 知识点2：详细分析跑马灯代码
- 知识点3：代码创建布局拓展
- 知识点4：通过XML创建布局实现跑马灯

任务挑战

完成如图效果

参考华为开发文档: <https://developer.harmonyos.com/cn/docs/documentation/doc-guides/ui-java-layout-dependentlayout-0000001050729536>



THANKS

更多学习视频，关注宅客学院.....

