# ConstraintLayout

<https://www.jianshu.com/p/193c6a906f20>

<https://developer.android.google.cn/training/constraint-layout?hl=zh_cn>

[https://www.bilibili.com/video/BV1f4411a7wh](https://www.bilibili.com/video/BV1f4411a7wh?from=search&seid=4439674350248442097)

<https://www.bilibili.com/video/BV1F4411Y7it>

<https://www.bilibili.com/video/BV1A4411u7Va>

# Android 中String的优雅用法

<https://blog.csdn.net/y874961524/article/details/57413814>

## String的一般使用

在资源文件目录res/values/新建一个strings.xml文件,这个文件中存放String的资源文件, 当然文件名没有规定,你可以随意起,但是根节点必须是<resource> , 资源节点必须是<string>

定义string资源:

<string name="app\_name">TextViewDemo</string>

Java代码中获取:

context.getString(R.string.app\_name);

xml文件中获取:

@string/app\_name

## string-array的使用

字符串数组是一种使用 name 属性（并非 XML 文件的名称）中提供的值进行引用的简单资源。 因此，您可以在一个 XML 文件中将字符串数组资源与其他简单资源合并在一起，放在 <resources> 元素之下。

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string-array name="string\_array\_name">

*<!--可以包含多个item-->*

<item>text\_string</item>

</string-array>

<string-array name="planets\_array">

<item>Mercury</item>

<item>Venus</item>

<item>Earth</item>

<item>Mars</item>

</string-array>

</resources>

Java代码中进行引用:

Resources res = getResources();

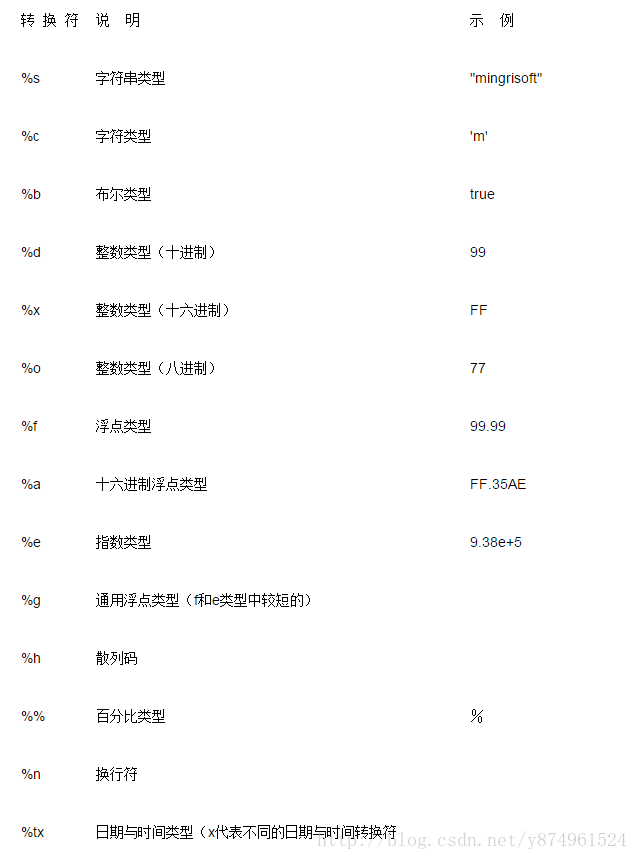
String[] planets = res.getStringArray(R.array.planets\_array);

## 格式化String资源

xml文件中定义

<string name="format">a first num %1$s , a second num %2$s</string>

1$ 和 2$ 表示第几个字符串的意思

%d是十进制的意思,类似的还有  


Java文件中进行引用

*// %1$s是字符串, %2$d是一个十进制数字*

*// String.format()会去除字符串中的样式信息*

mTvFormat.setText(String.format(getString(R.string.format), "one", "two"));



这里需要注意的是String.format()会将文本中的特殊字符进行转义,如果你在文本中添加了一些样式如<u> </u>, 这时候样式会失效,需要在之前调用TextUtils.htmlEncode() 将特殊字符转换成Html样式,或者在资源文件中直接写Html的特殊字符

**xml中将一些特殊符号转义成html格式**

<resources>

<string name="welcome\_messages">Hello, %1$s! You have &lt;b>%2$d new messages&lt;/b>.</string>

</resources>

Java中进行引用

Resources res = getResources();

String text = String.format(res.getString(R.string.welcome\_messages), username, mailCount);

*// 还原成样式文本*

CharSequence styledText = Html.fromHtml(text);

也可以使用TextUtils中的htmlEncode()方法进行自动转义

String escapedUsername = TextUtil.htmlEncode(username);

Resources res = getResources();

String text = String.format(res.getString(R.string.welcome\_messages), escapedUsername, mailCount);

CharSequence styledText = Html.fromHtml(text);

/\*\*

\* TextUtils中的htmlEncode(),将一些字符处理成html格式

\* Html-encode the string.

\* @param s the string to be encoded

\* @return the encoded string

\*/

public static String htmlEncode(String s) {

StringBuilder sb = new StringBuilder();

char c;

for (int i = 0; i < s.length(); i++) {

c = s.charAt(i);

switch (c) {

case '<':

sb.append("&lt;"); *//$NON-NLS-1$*

break;

case '>':

sb.append("&gt;"); *//$NON-NLS-1$*

break;

case '&':

sb.append("&amp;"); *//$NON-NLS-1$*

break;

case '\'':

*//http://www.w3.org/TR/xhtml1*

*// The named character reference &apos; (the apostrophe, U+0027) was introduced in*

*// XML 1.0 but does not appear in HTML. Authors should therefore use &#39; instead*

*// of &apos; to work as expected in HTML 4 user agents.*

sb.append("&#39;"); *//$NON-NLS-1$*

break;

case '"':

sb.append("&quot;"); *//$NON-NLS-1$*

break;

default:

sb.append(c);

}

}

return sb.toString();

}

## string资源中携带简单的html样式

string中可以直接放一下格式化的html样式  
如

* <b> 表示粗体文本。
* <i> 表示斜体文本。
* <u> 表示 下划线 文本。

这个时候如果不使用String.format()可以直接使用,使用String.format()需要进行html转义,就像上面两种情况那样

## 使用CDATA实现复杂样式

CDATA标记中的文本不会被XML解析器解析,所以就不需要转义

<string name="colors">

<![CDATA[

<font color=red>红</font>

<font color=\'#ff7f00\'>橙</font>

<font color=yellow>黄</font>

<font color=green>绿</font>

<font color=cyan>青</font>

<font color=blue>蓝</font>

<font color=purple>紫</font>

]]>

</string>

Java中进行使用

mTvColor.setText(Html.fromHtml(getString(R.string.colors)))*;*

效果  


# ClearEditText

<https://www.jianshu.com/p/6b2b0f6873db>

<com.johnzero.passwordbook.view.ClearEditText  
 android:id="@+id/et\_search"  
 android:layout\_width="0dp"  
 android:layout\_weight="3"  
 android:layout\_height="match\_parent"  
 android:hint="搜索标题"  
 android:background="#ffffff"  
 android:drawableRight="@mipmap/clear"/>

