#### startsWith(String str)方法

检查形参参数是否与你要检查的字符串开头相同，返回的是布尔值，形参参数可以是字符，也可以是字符串。

startsWith(String str, int Index)则是从你要比较的原字符串的指定下标开始和形参作比较

这里边呢str.startsWith(String str)和str.startsWith(string str,0)是没有区别的

#### substring(int beginIndex)方法

截取字符串，

#### String.format()——字符串格式化

在开发的时候一段字符串的中间某一部分是需要可变的 比如一个Textview需要显示”XXX用户来自 上海 年龄 21 性别 男”

其中的 XXX 是用户名 每个用户也是不一样的

地区 上海 为可变的string数据

年龄 21 为可变的int数据

性别 男 为可变的string数据

遇到这种情况你们是怎么样解决的呢?把这段字符串保存在常量类里吗?不!我们应该遵循Google的开发模式

String userName="XXX";

String userProvince="上海";

int userAge=21;

String userSex="男";

String string=getResources().getString(R.string.user\_info);

String userInfo=String.format(string,userName,userProvince,userAge,userSex);

String.format()字符串常规类型格式化的两种重载方式：

format(String format, Object… args) 新字符串使用本地语言环境，制定字符串格式和参数生成格式化的新字符串。

format(Locale locale, String format, Object… args) 使用指定的语言环境，制定字符串格式和参数生成格式化的字符串。

常规类型的格式化：

|  |  |  |
| --- | --- | --- |
| **转换符** | **详细说明** | **示例** |
| %s | 字符串类型 | “喜欢请收藏” |
| %c | 字符类型 | ‘m’ |
| %b | 布尔类型 | true |
| %d | 整数类型（十进制） | 88 |
| %x | 整数类型（十六进制） | FF |
| %o | 整数类型（八进制） | 77 |
| %f | 浮点类型 | 8.888 |
| %a | 十六进制浮点类型 | FF.35AE |
| %e | 指数类型 | 9.38e+5 |
| %g | 通用浮点类型（f和e类型中较短的） | 不举例(基本用不到) |
| %h | 散列码 | 不举例(基本用不到) |
| %% | 百分比类型 | ％(%特殊字符%%才能显示%) |
| %n | 换行符 | 不举例(基本用不到) |
| %tx | 日期与时间类型（x代表不同的日期与时间转换符) | 不举例(基本用不到) |

搭配转换的标志：

|  |  |  |  |
| --- | --- | --- | --- |
| **标志** | **说明** | **示例** | **结果** |
| + | 为正数或者负数添加符号 | (“%+d”,15) | +15 |
| 0 | 数字前面补0(加密常用) | (“%04d”, 99) | 0099 |
| 空格 | 在整数之前添加指定数量的空格 | (“% 4d”, 99) | 99 |
| , | 以“,”对数字分组(常用显示金额) | (“%,f”, 9999.99) | 9,999.990000 |
| ( | 使用括号包含负数 | (“%(f”, -99.99) | (99.990000) |
| # | 如果是浮点数则包含小数点，如果是16进制或8进制则添加0x或0 | (“%#x”, 99)(“%#o”, 99) | 0x63 0143 |
| < | 格式化前一个转换符所描述的参数 | (“%f和%<3.2f”, 99.45) | 99.450000和99.45 |
| d,%2$s”, 99,”abc”) | 99,abc |  |  |

日期和时间字符串格式化：

|  |  |  |
| --- | --- | --- |
| **标志** | **说明** | **示例** |
| c | 包括全部日期和时间信息 | 星期六 十月 27 14:21:20 CST 2007 |
| F | “年-月-日”格式 | 2007-10-27 |
| D | “月/日/年”格式 | 10/27/07 |
| r | “HH:MM:SS PM”格式（12时制） | 02:25:51 下午 |
| T | “HH:MM:SS”格式（24时制） | 14:28:16 |
| R | “HH:MM”格式（24时制） | 14:28 |