# 基本概念

# 源码

# 方法

## **char charAt(int index)**

返回指定索引处的 char 值。

## int compareTo(Object o)

把这个字符串和另一个对象比较。

## **String concat(String str)**

将指定字符串连接到此字符串的结尾。

## **boolean endsWith(String suffix)**

测试此字符串是否以指定的后缀结束。

## boolean equals(Object anObject)

将此字符串与指定的对象比较。

## **byte[] getBytes()**

使用平台的默认字符集将此 String 编码为 byte 序列，并将结果存储到一个新的 byte 数组中。

## **public static String join(CharSequence delimiter, CharSequence... elements)**

返回一个新 String，该字符串由 CharSequence 元素的副本与指定分隔符的副本连接在一起。

## **public static String join(CharSequence delimiter,Iterable<? extends CharSequence> elements)**

## **public String substring(int beginIndex)**

从字符串中提取子字符串，索引从0开始

返回从 beginIndex 开始到字符串末尾的子字符串

## **public String substring(int beginIndex, int endIndex)**

beginIndex：起始索引（包含）。

endIndex：结束索引（不包含）

返回从 beginIndex 开始到 endIndex - 1 的子字符串。

## **public String replace(CharSequence target, CharSequence replacement)**

返回一个新的字符串，其中所有 target 都被替换为 replacement，如果 target 未找到，则返回原字符串的副本。

## **public String replaceAll(String regex, String replacement)**

## **public int indexOf(String str)**

返回子字符串 str 在当前字符串中首次出现的索引（从 0 开始），如果未找到，返回 -1。

# 为什么 IDEA 建议去掉 StringBuilder，而要使用 “+” 拼接字符串

1、单纯的字符串拼接使用“+”，更快更简洁。

2、循环拼接时使用“+”拼接字符串效率较低，推荐使用 StringBuilder。