**String，StringBuilder，StringBuffer三者的区别**

**String：适用于少量的字符串操作的情况**

**StringBuilder：适用于单线程下在字符缓冲区进行大量操作的情况**

**StringBuffer：适用多线程下在字符缓冲区进行大量操作的情况**

这三个类之间的区别主要是在两个方面，即运行速度和线程安全这两方面。

**1.首先说运行速度，或者说是执行速度，在这方面运行速度快慢为：StringBuilder > StringBuffer > String**

**String最慢的原因：**

**String为字符串常量，而StringBuilder和StringBuffer均为字符串变量，即String对象一旦创建之后该对象是不可更改的，但后两者的对象是变量，是可以更改的。**以下面一段代码为例：

1 String str="abc";

2 System.out.println(str);

3 str=str+"de";

4 System.out.println(str);

　　如果运行这段代码会发现先输出“abc”，然后又输出“abcde”，好像是str这个对象被更改了，其实，这只是一种假象罢了，JVM对于这几行代码是这样处理的，首先创建一个String对象str，并把“abc”赋值给str，然后在第三行中，其实JVM又创建了一个新的对象也名为str，然后再把原来的str的值和“de”加起来再赋值给新的str，而原来的str就会被JVM的垃圾回收机制（GC）给回收掉了，所以，str实际上并没有被更改，也就是前面说的String对象一旦创建之后就不可更改了。所以，Java中对String对象进行的操作实际上是一个不断创建新的对象并且将旧的对象回收的一个过程，所以执行速度很慢。

　　而StringBuilder和StringBuffer的对象是变量，对变量进行操作就是直接对该对象进行更改，而不进行创建和回收的操作，所以速度要比String快很多。

　　另外，有时候我们会这样对字符串进行赋值

1 String str="abc"+"de";

2 StringBuilder stringBuilder=new StringBuilder().append("abc").append("de");

3 System.out.println(str);

4 System.out.println(stringBuilder.toString());

　　这样输出结果也是“abcde”和“abcde”，但是String的速度却比StringBuilder的反应速度要快很多，这是因为第1行中的操作和　　String str="abcde";　　是完全一样的，所以会很快，而如果写成下面这种形式

1 String str1="abc";

2 String str2="de";

3 String str=str1+str2;

那么JVM就会像上面说的那样，不断的创建、回收对象来进行这个操作了。速度就会很慢。

　　2. 再来说线程安全

**在线程安全上，StringBuilder是线程不安全的，而StringBuffer是线程安全的**

　　如果一个StringBuffer对象在字符串缓冲区被多个线程使用时，StringBuffer中很多方法可以带有synchronized关键字，所以可以保证线程是安全的，但StringBuilder的方法则没有该关键字，所以不能保证线程安全，有可能会出现一些错误的操作。所以如果要进行的操作是多线程的，那么就要使用StringBuffer，但是在单线程的情况下，还是建议使用速度比较快的StringBuilder。