字符串

字符串

- 用双引号括起来的0个或多个字符就是一个字符串字面量
  - “hello”
  - “1”
  - “”

字符串变量

- String s;
- String是一个类，String的变量是对象的管理者而非所有者
  - 就像数组变量是数组的管理者而非所有者一样

new＝创建

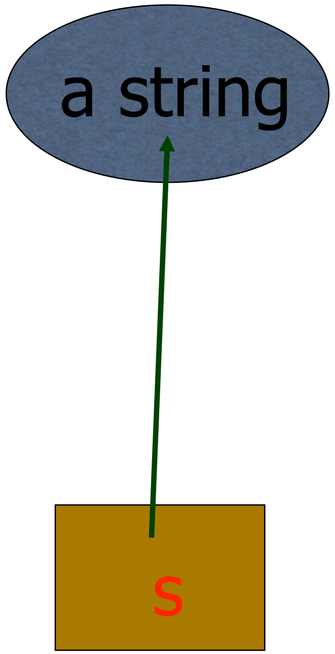
```
String s = new String(“a string”);
```

创建了一个String的对象

用“a string”初始化这个对象

创建管理这个对象的变量s

让s管理这个对象



初始化字符串变量

- String s = “hello”;
- 编译器帮你创建一个String类的对象交给s来管理

字符串连接

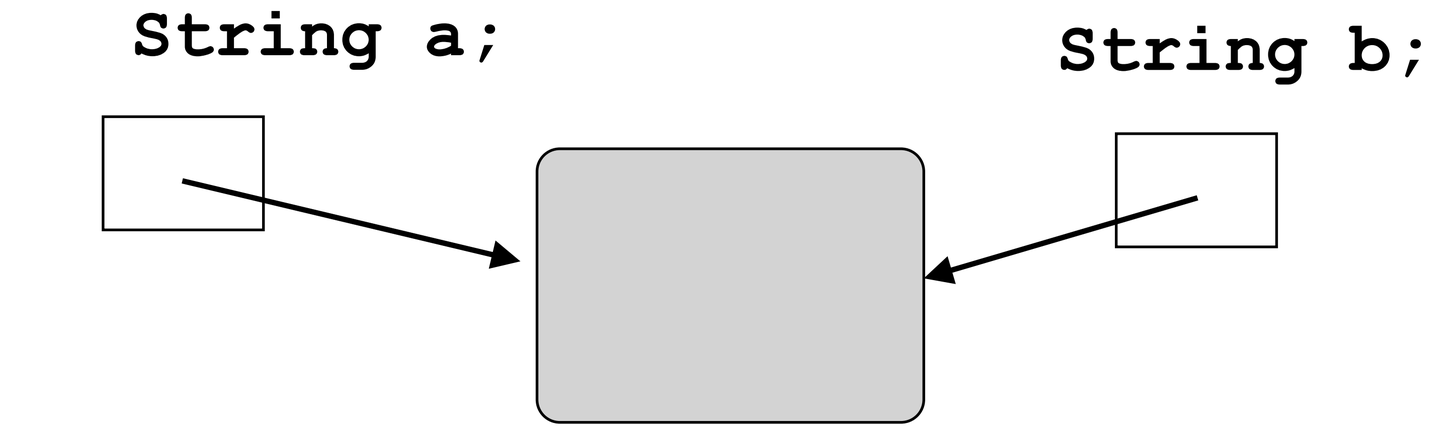
- 用加号（+）可以连接两个字符串
  - “hello”+”world”—>”helloworld”
- 当这个+的一边是字符串而另一边不是时，会将另一边表达为字符串然后做连接
  - “I’m ”+18—>”I’m 18”
  - 1+2+”age”—>”3age”
  - “age”+1+2—>”age12”

输入字符串

- in.next();读入一个单词，单词的标志是空格
  - 空格包括空格、tab和换行
- in.nextLine();读入一整行

字符串变量

对象变量的赋值



int a;

32

int b;

32

比较两个String

```
if(input == "bye") {
    ...
}
```

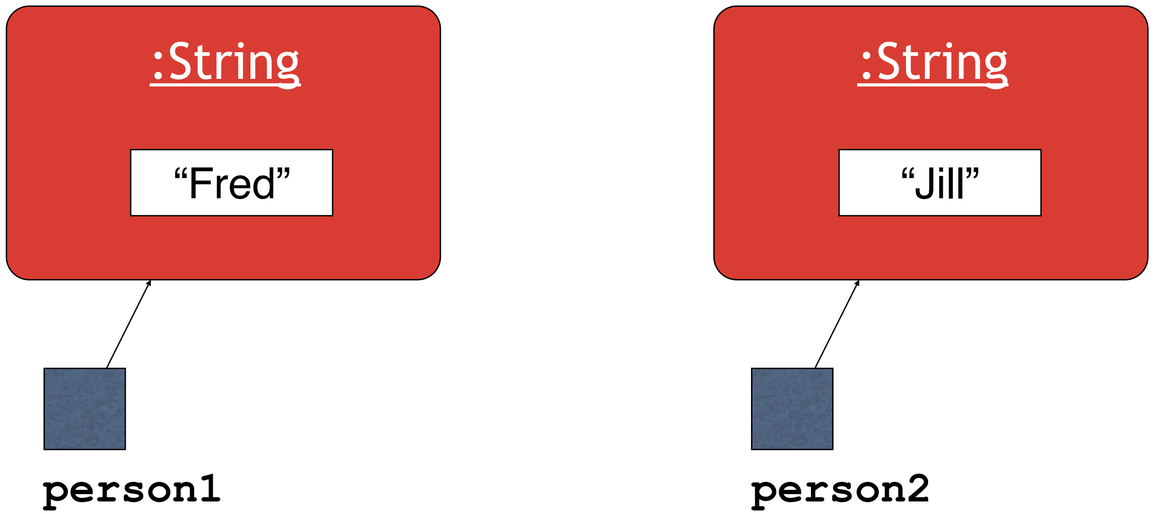
比较是否同一个

```
if(input.equals("bye")) {
    ...
}
```

比较内容是否相同

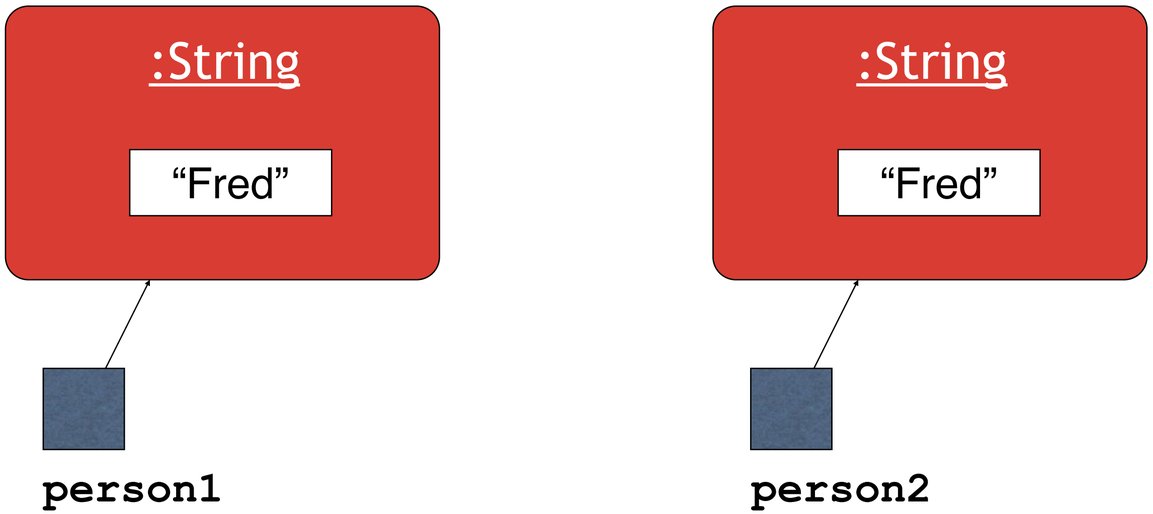
- Strings应该用 **`.equals`** 来比较

同一个还是相同



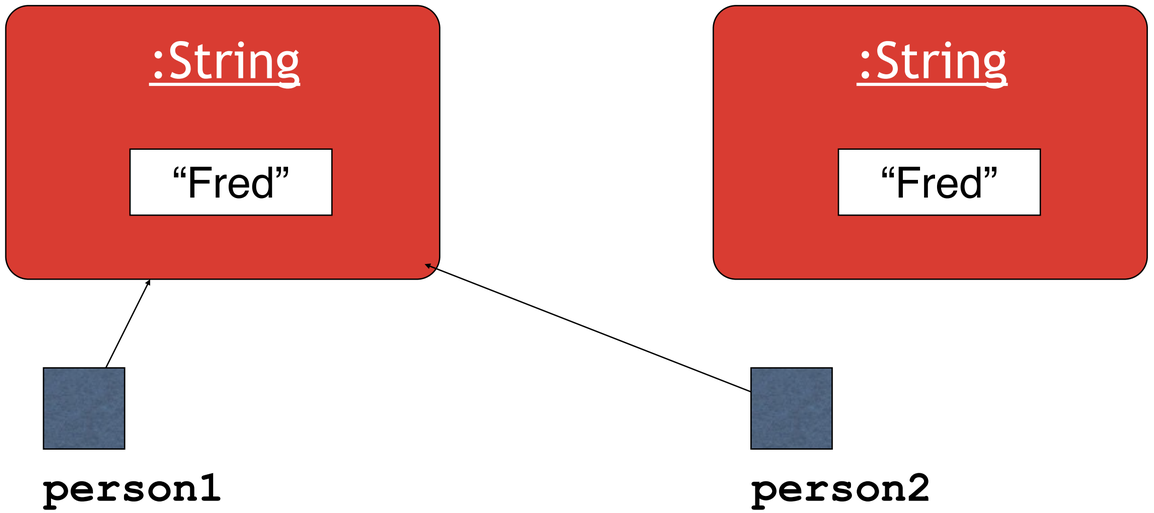

```
person1 == person2 ?
```

同一个还是相同




```
person1 == person2 ?
```

同一个还是相同

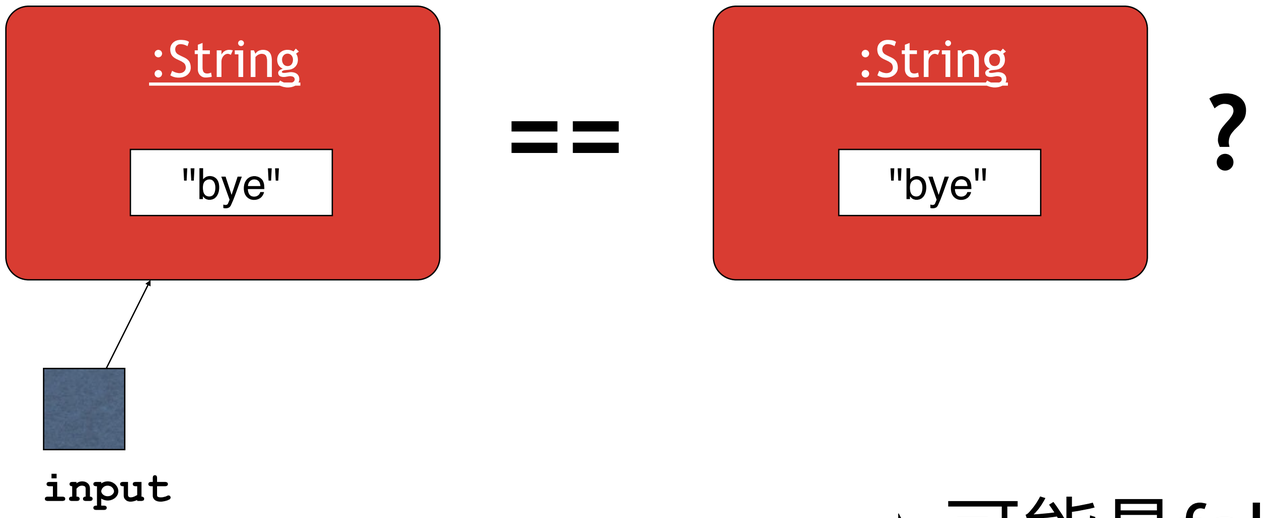



```
person1 == person2 ?
```

同一个还是相同

```
String input = in.next();
if(input == "bye") {
    ...
}
```

== 比较是否同一个

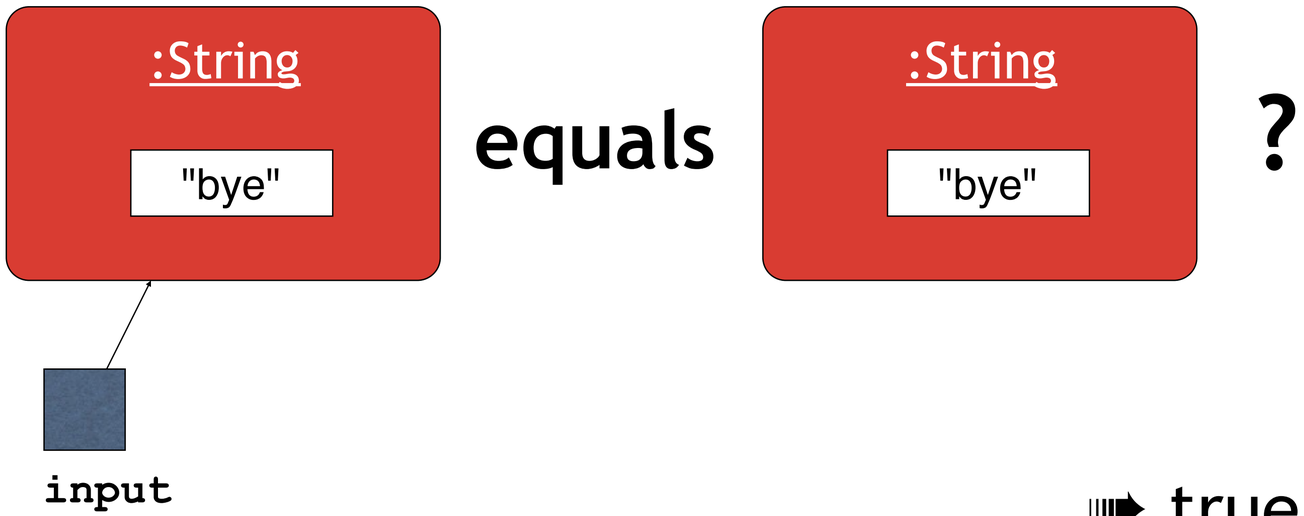


➠ 可能是false!

同一个还是相同

```
String input = reader.getInput();
if(input.equals("bye")) {
    ...
}
```

equals 比较
内容是否相同



➠ true!

字符串运算

字符串操作

- 字符串是对象，对它的所有操作都是通过“.”这个运算符来进行的
  - 字符串.操作
- 它表示对.左边的这个字符串做右边的那个操作
- 这里的字符串可以是变量也可以是常量

Strings大小的比较

- 两个字符串可以比较大小:

```
s1.compareTo(s2)
```

如果s1比s2小，那么结果是负的；如果s1和s2相等，那么结果是0；如果s1比s2大，那么结果是正的

- `compareToIgnoreCase`可以不区分大小写地来比较大小

获得String的长度

- 用length() 函数

```
String name = “Hellola”,
str1 = “one”,
str2 = “”,
str3;
```

```
name.length( );
```
→ **7**

```
str1.length( );
```
→ **3**

```
str2.length( );
```
→ **0**

```
str3.length( );
```
→ **Error!**

错误是因为str3没有管理任何String对象

访问String里的字符

- `s.charAt(index)`
  - 返回在*index*上的单个字符
  - *index*的范围是0到length( ) - 1.
  - 第一个字符的*index*是0，和数组一样
- 但是不能用*for-each*循环来遍历字符串

得到子串

- s.substring(n)
  - 得到从n号位置到末尾的全部内容
- s.substring(b,e)
  - 得到从b号位置到e号位置之前的内容

寻找字符

- `s.indexOf(c)`
  - 得到c字符所在的位置，-1表示不存在
- `s.indexOf(c, n)`
  - 从n号位置开始寻找c字符
- `s.indexOf(t)`
  - 找到字符串t所在的位置
- 从右边开始找
  - `s.lastIndexOf(c)`
  - `s.lastIndexOf(c, n)`
  - `s.lastIndexOf(t)`

其他String操作

- s.startsWith(t)
- s.endsWith(t)
- s.trim()
- s.replace(c1,c2)
- s.toLowerCase()
- s.toUpperCase()

不可变的String

- 所有的字符串都是不可变的，对它们的操作的结果都是制造新的字符串出来

  String s = “abc ”;

  System.out.println(s.toUpperCase());

  System.out.println(s);

在switch-case中使用字符串

```
switch ( s ) {
case “this”:...break;
case “that”:...break;
}
```