

Scala 字符串

以下实例将字符串赋值给一个常量：

```
object Test {  
  val greeting: String = "Hello,World!"  
  
  def main(args: Array[String]) {  
    println( greeting )  
  }  
}
```

以上实例定义了变量 `greeting`，为字符串常量，它的类型为 `String` (`java.lang.String`)。

在 Scala 中，字符串的类型实际上是 Java `String`，它本身没有 `String` 类。

在 Scala 中，`String` 是一个不可变的对象，所以该对象不可被修改。这就意味着你如果修改字符串就会产生一个新的字符串对象。

但其他对象，如数组就是可变的对象。接下来我们会为大家介绍常用的 `java.lang.String` 方法。

创建字符串

创建字符串实例如下：

```
var greeting = "Hello World!";  
  
或  
  
var greeting:String = "Hello World!";
```

你不一定为字符串指定 `String` 类型，因为 Scala 编译器会自动推断出字符串的类型为 `String`。

当然我们也可以直接显示的声明字符串为 `String` 类型，如下实例：

```
object Test {  
  val greeting: String = "Hello, World!"  
  
  def main(args: Array[String]) {  
    println( greeting )  
  }  
}
```

执行以上代码，输出结果为：

```
$ scalac Test.scala
$ scala Test
Hello, world!
```

我们前面提到过 String 对象是不可变的，如果你需要创建一个可以修改的字符串，可以使用 String Builder 类，如下实例：

```
object Test {
  def main(args: Array[String]) {
    val buf = new StringBuilder;
    buf += 'a'
    buf += "bcdef"
    println( "buf is : " + buf.toString );
  }
}
```

运行实例 »

执行以上代码，输出结果为：

```
$ scalac Test.scala
$ scala Test
buf is : abcdef
```

字符串长度

我们可以使用 length() 方法来获取字符串长度：

```
object Test {
  def main(args: Array[String]) {
    var palindrome = "www.runoob.com";
    var len = palindrome.length();
    println( "String Length is : " + len );
  }
}
```

执行以上代码，输出结果为：

```
$ scalac Test.scala
$ scala Test
String Length is : 14
```

字符串连接

String 类中使用 concat() 方法来连接两个字符串：

```
string1.concat(string2);
```

实例演示：

```
scala> "菜鸟教程官网：".concat("www.runoob.com");  
res0: String = 菜鸟教程官网： www.runoob.com
```

同样你也可以使用加号(+)来连接：

```
scala> "菜鸟教程官网： " + " www.runoob.com"  
res1: String = 菜鸟教程官网：  www.runoob.com
```

让我们看个完整实例：

```
object Test {  
  def main(args: Array[String]) {  
    var str1 = "菜鸟教程官网：";  
    var str2 = "www.runoob.com";  
    var str3 = "菜鸟教程的 Slogan 为：";  
    var str4 = "学的不仅是技术，更是梦想！";  
    println( str1 + str2 );  
    println( str3.concat(str4) );  
  }  
}
```

执行以上代码，输出结果为：

```
$ scalac Test.scala  
$ scala Test  
菜鸟教程官网：www.runoob.com  
菜鸟教程的 Slogan 为：学的不仅是技术，更是梦想！
```

创建格式化字符串

String 类中你可以使用 printf() 方法来格式化字符串并输出，String format() 方法可以返回 String 对象而不是 PrintStream 对象。以下实例演示了 printf() 方法的使用：

```
object Test {  
  def main(args: Array[String]) {  
    var floatVar = 12.456  
    var intVar = 2000
```

```
var stringVar = "菜鸟教程!"
var fs = printf("浮点型变量为 " +
               "%f, 整型变量为 %d, 字符串为 " +
               "%s", floatVar, intVar, stringVar)
println(fs)
}
```

执行以上代码，输出结果为：

```
$ scalac Test.scala
$ scala Test
浮点型变量为 12.456000, 整型变量为 2000, 字符串为  菜鸟教程!()
```

String 方法

下表列出了 java.lang.String 中常用的方法，你可以在 Scala 中使用：

序号	方法及描述
1	char charAt(int index) 返回指定位置的字符
2	int compareTo(Object o) 比较字符串与对象
3	int compareTo(String anotherString) 按字典顺序比较两个字符串
4	int compareToIgnoreCase(String str) 按字典顺序比较两个字符串，不考虑大小写
5	String concat(String str) 将指定字符串连接到此字符串的结尾
6	boolean contentEquals(StringBuffer sb) 将此字符串与指定的 StringBuffer 比较。
7	static String copyValueOf(char[] data) 返回指定数组中表示该字符序列的 String
8	static String copyValueOf(char[] data, int offset, int count) 返回指定数组中表示该字符序列的 String
9	boolean endsWith(String suffix)

	测试此字符串是否以指定的后缀结束
10	boolean equals(Object anObject) 将此字符串与指定的对象比较
11	boolean equalsIgnoreCase(String anotherString) 将此 String 与另一个 String 比较，不考虑大小写
12	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
13	byte[] getBytes(String charsetName) 使用指定的字符集将此 String 编码为 byte 序列，并将结果存储到一个新的 byte 数组中
14	void getChars(int srcBegin, int srcEnd, char[] dst, int dstBegin) 将字符从此字符串复制到目标字符数组
15	int hashCode() 返回此字符串的哈希码
16	int indexOf(int ch) 返回指定字符在此字符串中第一次出现处的索引
17	int indexOf(int ch, int fromIndex) 返回在此字符串中第一次出现指定字符处的索引，从指定的索引开始搜索
18	int indexOf(String str) 返回指定子字符串在此字符串中第一次出现处的索引
19	int indexOf(String str, int fromIndex) 返回指定子字符串在此字符串中第一次出现处的索引，从指定的索引开始
20	String intern() 返回字符串对象的规范化表示形式
21	int lastIndexOf(int ch) 返回指定字符在此字符串中最后一次出现处的索引
22	int lastIndexOf(int ch, int fromIndex) 返回指定字符在此字符串中最后一次出现处的索引，从指定的索引处开始进行反向搜索
23	int lastIndexOf(String str) 返回指定子字符串在此字符串中最右边出现处的索引

24	int lastIndexOf(String str, int fromIndex) 返回指定子字符串在此字符串中最后一次出现处的索引，从指定的索引开始反向搜索
25	int length() 返回此字符串的长度
26	boolean matches(String regex) 告知此字符串是否匹配给定的正则表达式
27	boolean regionMatches(boolean ignoreCase, int toffset, String other, int ooffset, int len) 测试两个字符串区域是否相等
28	boolean regionMatches(int toffset, String other, int ooffset, int len) 测试两个字符串区域是否相等
29	String replace(char oldChar, char newChar) 返回一个新的字符串，它是通过用 newChar 替换此字符串中出现的所有 oldChar 得到的
30	String replaceAll(String regex, String replacement) 使用给定的 replacement 替换此字符串所有匹配给定的正则表达式的子字符串
31	String replaceFirst(String regex, String replacement) 使用给定的 replacement 替换此字符串匹配给定的正则表达式的第一个子字符串
32	String[] split(String regex) 根据给定正则表达式的匹配拆分此字符串
33	String[] split(String regex, int limit) 根据匹配给定的正则表达式来拆分此字符串
34	boolean startsWith(String prefix) 测试此字符串是否以指定的前缀开始
35	boolean startsWith(String prefix, int toffset) 测试此字符串从指定索引开始的子字符串是否以指定前缀开始。
36	CharSequence subSequence(int beginIndex, int endIndex) 返回一个新的字符序列，它是此序列的一个子序列
37	String substring(int beginIndex) 返回一个新的字符串，它是此字符串的一个子字符串
38	String substring(int beginIndex, int endIndex)

	返回一个新字符串，它是此字符串的一个子字符串
39	char[] toCharArray() 将此字符串转换为一个新的字符数组
40	String toLowerCase() 使用默认语言环境的规则将此 String 中的所有字符都转换为小写
41	String toLowerCase(Locale locale) 使用给定 Locale 的规则将此 String 中的所有字符都转换为小写
42	String toString() 返回此对象本身（它已经是一个字符串！）
43	String toUpperCase() 使用默认语言环境的规则将此 String 中的所有字符都转换为大写
44	String toUpperCase(Locale locale) 使用给定 Locale 的规则将此 String 中的所有字符都转换为大写
45	String trim() 删除指定字符串的首尾空白符
46	static String valueOf(primitive data type x) 返回指定类型参数的字符串表示形式

✍ 点我分享笔记