

Scala String

In scala, string is a combination of characters or we can say it is a sequence of characters. It is index based data structure and use linear approach to store data into memory. String is immutable in scala like java. You can manipulate string and can apply method to get desired result but you can't change original string object.

Scala String Example

This is a simple string example in which we have created a string variable.

```
class StringExample{  
    var s1 = "Scala string example"  
  
    def show(){  
        println(s1)  
    }  
}  
  
object MainObject{  
    def main(args:Array[String]){  
        var s = new StringExample()  
        s.show()  
    }  
}
```

Output:

Scala string example

Scala Immutable String Example 1

In this example, we are creating a string variable and trying to modify string object. Let's see what happens.

```
class StringExample{
```

```

var s1 = "Scala string example"

"This is "+s1      // Modifying string object

def show(){
    println(s1)
}

}

object MainObject{

    def main(args:Array[String]){

        var s = new StringExample()

        s.show()

    }

}

```

Output:

Scala string example

Scala Immutable String Example 2

In this example, we are modifying the string and assigning it to a variable so that we can keep reference of modified string. Here, a new string object is created after modification. The s1 variable now refers to new string object. The old string object is still available in the memory but does not have any reference.

```

class StringExample{

    var s1 = "Scala string example"

    s1 = "This is "+s1

    def show(){

        println(s1)

    }

}

```

```

}

object MainObject{

  def main(args:Array[String]){

    var s = new StringExample()

    s.show()

  }

}

```

Output:

This is Scala string example

Scala String Comparison Example

In scala, you can compare two string objects by using == (equal) method. The following program describes how to use equal operator. It returns boolean value either true or false.

```

class StringExample{

  var s1 = "Scala string example"

  var s2 = "Hello Scala"

  var s3 = "Hello Scala"

  def show(){

    println(s1 == s2)

    println(s2 == s3)

  }

}

object MainObject{

  def main(args:Array[String]){

    var s = new StringExample()

    s.show()

```

```
}  
}
```

Output:

false

true

Scala String Methods

Scala String equals() Method Example

You can also use `equal()` method to compare two string objects. It returns true if both string object are equal otherwise returns false.

```
class StringExample{  
    var s1 = "Scala string example"  
    var s2 = "Hello Scala"  
    var s3 = "Hello Scala"  
    def show(){  
        println(s1.equals(s2))  
        println(s2.equals(s3))  
    }  
}  
  
object MainObject{  
    def main(args:Array[String]){  
        var s = new StringExample()  
        s.show()  
    }  
}
```

```
}
```

Output:

false

true

Scala compareTo() Method Example

The compareTo() method compares the given string with current string lexicographically. It returns positive number, negative number or 0.

If first string is greater than second string, it returns positive number (difference of character value). If first string is less than second string, it returns negative number and if first string is equal to second string, it returns 0.

s1 > s2 => positive number

s1 < s2 => negative number

s1 == s2 => 0

```
class StringExample{  
    var s1 = "Scala string example"  
    var s2 = "Hello Scala"  
    var s3 = "Hello Scala"  
    def show(){  
        println(s1.compareTo(s2))  
        println(s2.compareTo(s3))  
    }  
}  
  
object MainObject{  
    def main(args:Array[String]){  
        var s = new StringExample()  
        s.show()  
    }  
}
```

```
}
```

Output:

11

0

Scala Concatenation Example by using + (Plus) Operator

In scala, you can concatenate string by using + operator. An example is given below in which two strings are concatenated by using + operator.

```
class StringExample{  
    var s1 = "This is "+"Scala string example"  
    def show(){  
        println(s1)  
    }  
}  
  
object MainObject{  
    def main(args:Array[String]){  
        var s = new StringExample()  
        s.show()  
    }  
}
```

Output:

This is Scala string example

Scala String concat() Method Example

Apart from + operator, you can also use concat() method to concatenate two strings. It is a predefined method of string class.

```
class StringExample{  
    var s1 = "This is "
```

```

var s2 = "Scala string example"

def show(){

    println(s1.concat(s2))

}

}

object MainObject{

    def main(args:Array[String]){

        var s = new StringExample()

        s.show()

    }

}

```

Output:

This is Scala string example

Scala substring() Method Example

The substring() method is used to get substring from a string. By specifying start and end index as argument you can get substring according to your requirement. It is a predefined method of string class.

```

class StringExample3{

    var s1 = "Scala string example"

    def show(){

        println(s1.substring(0,5))

    }

}

object MainObject{

    def main(args:Array[String]){

        var s = new StringExample3()

        s.show()

    }

}

```

```
}
```

```
}
```

Output:

Scala