

Die Klasse String in



```
public class KlasseString {  
  
    public static void main (String args[]) {  
  
        String s = "Hallo";  
        //bisher bekannte, vereinfachte Notation – sieht aus wie ein primitiver Datentyp!  
  
        System.out.println("String in Kurzform deklariert: " + s);  
  
        String t = new String("Hallo");  
        //ein neues Objekt der Klasse String wird erzeugt, was anhand des „new“-  
        Operators erkannt werden kann (zum Klassen-Objekt-Konzept später mehr!)  
  
        System.out.println("String original deklariert: " + t);  
  
        //mit „String t – Zeile“ gleichzusetzen:  
        System.out.print("String als array von char: ");  
        char[] tester = {'H','a','l','l','o'};  
  
        //Ausgabe des Buchstaben-arrays – man erhält dasselbe wie bei der Ausgabe von t!  
        for (int i=0; i<=4; i++) System.out.print(tester[i]);  
  
        System.out.println(); System.out.println();  
    }  
}
```

```
String in Kurzform deklariert: Hallo  
String original deklariert: Hallo  
String als array von char: Hallo
```

Wie oben erkennbar ist ein String in Java technisch gesehen eine Aneinanderreihung von Einzelbuchstaben (vgl. Datentyp char – ein einzelner Buchstabe).

String ist darüber hinaus – im Gegensatz zu den bisher bekannten (einfachen) Datentypen int oder double oder char – **eine Klasse** und wird gerade deshalb auch groß geschrieben!

Die Klasse String beinhaltet eine Vielzahl von Methoden, mit denen man Zeichenketten auf vielfältige Weise manipulieren und die man alle in der offiziellen Dokumentation nachlesen kann.

Im Folgenden soll jedoch wenigstens ein kleiner Ausschnitt aus dieser Menge von Funktionen aufgelistet werden, deren genauere Wirkung Du Dir dann im Nachhinein in Form kleiner Programme selbst näherbringen kannst!

String-Methoden

Ausschnitt/ Mini-Beispiele

char **charAt**(int index) – Returns the char value at the specified index.

```
String s = new String("Test");  
System.out.println("1. Buchstabe von s: " + s.charAt(0));
```

1. Buchstabe von s: T

String **concat**(String str) – Concatenates the specified string to the end of this string.

```
String s = new String("Test");  
s = s.concat(" Nummer 1");  
System.out.println(s);
```

Test Nummer 1

boolean **endsWith**(String suffix) – Tests if this string ends with the specified suffix.

```
String s = new String("Test");  
if (s.endsWith("st")) { System.out.println("ja");}
```

ja

boolean **equals**(Object anObject) – Compares this string to the specified object.

```
String s = new String("Test"); String t = "Test";  
if (s == t) System.out.println("reicht"); else System.out.println("reicht nicht");  
if (s.equals(t)) System.out.println("ja");
```

reicht nicht
ja

int **indexOf**(char ch) – Returns the index within this string of the first occurrence of the specified character.

```
String s = new String("Tester");  
System.out.println(s.indexOf('e'));
```

1

int **length**() – Returns the length of this string.

```
String s = new String("Tester");  
System.out.println(s.length());
```

6

String **substring**(int beginIndex, int endIndex) – Returns a new string that is a substring of this string.

```
String s = new String("Testermania");  
System.out.println(s.substring(3,7));
```

term

String **toUpperCase**() – Converts all of the characters in this String to upper case using the rules of the default locale.

```
String s = new String("Testermania");  
System.out.println(s.toUpperCase());
```

TESTERMANIA