Java String format()

The **java string format()** method returns the formatted string by given locale, format and arguments.

If you don't specify the locale in String.format() method, it uses default locale by calling *Locale.getDefault()* method.

The format() method of java language is like *sprintf()* function in c language and *printf()* method of java language.

Internal implementation

1. **public** **static** String format(String format, Object... args) {
2. **return** **new** Formatter().format(format, args).toString();
3. }

Signature

There are two type of string format() method:

1. **public** **static** String format(String format, Object... args)
2. and,
3. **public** **static** String format(Locale locale, String format, Object... args)

Parameters

**locale** : specifies the locale to be applied on the format() method.

**format** : format of the string.

**args** : arguments for the format string. It may be zero or more.

Returns

formatted string

Throws

**NullPointerException** : if format is null.

**IllegalFormatException** : if format is illegal or incompatible.

Java String format() method example

1. **public** **class** FormatExample{
2. **public** **static** **void** main(String args[]){
3. String name="sonoo";
4. String sf1=String.format("name is %s",name);
5. String sf2=String.format("value is %f",32.33434);
6. String sf3=String.format("value is %32.12f",32.33434);//returns 12 char fractional part filling with 0
8. System.out.println(sf1);
9. System.out.println(sf2);
10. System.out.println(sf3);
11. }}

[**Test it Now**](https://www.javatpoint.com/opr/test.jsp?filename=FormatExample)

name is sonoo

value is 32.334340

value is 32.334340000000

Java String Format Specifiers

Here, we are providing a table of format specifiers supported by the Java String.

|  |  |  |
| --- | --- | --- |
| **Format Specifier** | **Data Type** | **Output** |
| %a | floating point (except *BigDecimal*) | Returns Hex output of floating point number. |
| %b | Any type | "true" if non-null, "false" if null |
| %c | Character | Unicode character |
| %d | integer (incl. byte, short, int, long, bigint) | Decimal Integer |
| %e | floating point | decimal number in scientific notation |
| %f | floating point | decimal number |
| %g | floating point | decimal number, possibly in scientific notation depending on the precision and value. |
| %h | any type | Hex String of value from hashCode() method. |
| %n | None | Platform-specific line separator. |
| %o | integer (incl. byte, short, int, long, bigint) | Octal number |
| %s | any type | String value |
| %t | Date/Time (incl. long, Calendar, Date and TemporalAccessor) | %t is the prefix for Date/Time conversions. More formatting flags are needed after this. See Date/Time conversion below. |
| %x | integer (incl. byte, short, int, long, bigint) | Hex string. |

Java String format() Method Example 2

This method supports various data types and formats them into a string type. Let us see an example.

1. **public** **class** FormatExample2 {
2. **public** **static** **void** main(String[] args) {
3. String str1 = String.format("%d", 101);          // Integer value
4. String str2 = String.format("%s", "Amar Singh"); // String value
5. String str3 = String.format("%f", 101.00);       // Float value
6. String str4 = String.format("%x", 101);          // Hexadecimal value
7. String str5 = String.format("%c", 'c');          // Char value
8. System.out.println(str1);
9. System.out.println(str2);
10. System.out.println(str3);
11. System.out.println(str4);
12. System.out.println(str5);
13. }
15. }

[**Test it Now**](https://www.javatpoint.com/opr/test.jsp?filename=FormatExample2)

101

Amar Singh

101.000000

65

c

Java String format() Method Example 3

Apart from formatting, we can set width, padding etc. of any value. Let us see an example where we are setting width and padding for an integer value.

1. **public** **class** FormatExample3 {
2. **public** **static** **void** main(String[] args) {
3. String str1 = String.format("%d", 101);
4. String str2 = String.format("|%10d|", 101);  // Specifying length of integer
5. String str3 = String.format("|%-10d|", 101); // Left-justifying within the specified width
6. String str4 = String.format("|% d|", 101);
7. String str5 = String.format("|%010d|", 101); // Filling with zeroes
8. System.out.println(str1);
9. System.out.println(str2);
10. System.out.println(str3);
11. System.out.println(str4);
12. System.out.println(str5);
13. }
14. }