Printing with **System.out.format() or System.out.printf()**

**System.out.format("Print this number: %d", 25);**

**//Output:**

**//Print this number: 25**

In this example, a special format specifier %d is used in the first argument. The second argument is the integer value 25. The format() (or printf()) method will substitute the %d specifier with the value 25. The "%d" format specifier is used to print integer numbers. Integers are formatted with no decimal places.

After you specify the String you want to output, you then specify the value(s) you want formatted. Separate each argument, including the string, with commas. In the above example, the %d in the string will be replaced by the value 25. You can print several values as well.

**System.out.format("The numbers are %d and %d", 25, 700);**

**//Output:**

**// The numbers are 25 and 700**

If you want to print a decimal value, you can use the %f specifier, which will print a decimal value with 6 digits after the decimal. For example:

**System.out.format("This is a floating point value: %f", 2.345);**

**//Output:**

**// This is a floating point value: 2.345000**

You can also format more than one value by including more than one format specifier. You must make sure you include an argument for each specifier:

**System.out.format("The result of %f times %f is %f.", .25, 1.1, (.25 \* 1.1));**

**//Output:**

**// The result of 0.250000 times 1.100000 is 0.275000**

The values and format specifiers match by the order in which they appear. In the example above, the first %f will be the formatted value .25, the second %f will be the formatted value 1.1, and the last %f will be the formatted result of (.25 \* 1.1).

You can also specify how many digits after the decimal you'd like to display. Between the % and the "f" of the floating point format specifier, you can include information about the number of spaces a value should take up, and the number of digits that should appear after the decimal. For example:

**System.out.format("This is a nicer decimal value: %4.2f.",(.25 \* 1.1));**

**//Output:**

**//This is a nicer decimal value: 0.28.**

The format specifier "%4.2f" indicates that a floating point number is displayed in a total of 4 character spaces, including 2 digits after the decimal.

What happens if you try it this way:

**System.out.format("This is a nicer decimal value: %10.2f.",(.25 \* 1.1));**

**//Output:**

**//This is a nicer decimal value: 0.28.**

If the first number in the format specifier is more space than you need, extra padding will be added to the left of the value to make up the number of character spaces specified. In our example, we specified 10 spaces, but we only required 4, so 6 extra spaces were added. This can come in handy when you want to line up columns of decimal values:

**System.out.println("Some Values:");**

**System.out.format("%9.2f\n", 123.45);**

**System.out.format("%9.2f\n", .12);**

**System.out.format("%9.2f\n", 1.2);**

Try this code! You'll get the following output:

**Some Values:**

**123.45**

**0.12**

**1.20**

\n is a new line. If format %n is also used for the new line.

If you want to make your formatted values left-justified instead of right-justified, add a "-" (minus sign or dash) in front of the total space value:

**System.out.println("Some Values:");**

**System.out.format("Value 1: %-9.2f\n", 123.45);**

**System.out.format("Value 2: %-9.2f\n", .12);**

**System.out.format("Value 3: %-9.2f\n", 1.2);**

Output:

**Value 1: 123.45**

**Value 2: 0.12**

**Value 3: 1.20**

%10s is used for strings

**Exercise 1**  What would be the output of the following code segment?

**System.out.format (**"**Value:%7.2f%n**"**, 12.2);**

**System.out.format (**"**Another value:%3.2f%n**"**, 456.789);**

**System.out.format(**"**%d + %d is %d%n**"**, 5, 3, (5+3));**

**System.out.format (**"**The value %-5.1f is numeric.%n**"**, 5.0);**

**System.out.format (**"**The value %10s is a String%n**"**,** "**Hello**"**);**

**System.out.format (**"**The value %-10s is also a String%n**"**,** "**Bye**"**);**