

## SPRINT 12

# Output Formatting

What a value is and how we display it are two different things. We can adapt how we display various value literals through formatters that convert and display values in different ways.

To do this, we will use the `printf()` method that will contain the information we want to display, but includes special codes that change the way the values are shown on the screen.

## Decimal Formatters

The most basic is when we work with decimals. Sometimes, we need to display more digits in a number than are needed.

Here's the code for how this works.

### ***Listing 12-1.*** Decimal formatters

```
System.out.println(2.3f);  
System.out.printf("%.2f",2.3f);  
System.out.println();  
System.out.printf("%.4f",2.3f);  
System.out.println();
```

Here's how it works. You start with the `println()` statement, and then you create a string literal that holds the text you want to display.

Then, you use the percent sign to create a marker in the literal. Immediately following, you add the formatting code.

After the formatter string literal, you follow with a comma and the value you want to insert into the first place. You can insert more than one value in a string formatter, just separate them with commas.

The example in Listing 12-1 outputs the following:

```
2.3
2.30
2.3000
```

The `%.2f` code means that we are providing a floating-point number (a number with a decimal) with two digits in the decimal. This is defined using the `f` character in the code.

The last example uses `%.4f` that provides four digits in the output.

## Thousands Formatters

For a larger number, it is easier to read it if you use commas to separate each level of thousand in the number (or a period if you are in other parts of the world).

To do that, you add a comma in the formatter code. Also, if you are working with integers, or numbers without fractions, you use the character `d` in the formatter instead of `f`.

**Listing 12-2.** Thousands formatter

```
System.out.println(1000000);
System.out.printf("%,d",1000000);
System.out.println();
```

The output in this example is

```
1000000
1,000,000
```

## Currency Formatters

We can combine other text in with the formatter string and combine multiple formatters together. For example, what if we wanted to show a price? We should show that with two decimal places and with a thousands separator in the number.

**Listing 12-3.** Currency formatter

```

System.out.println(1000.2f);
System.out.printf("%.2f", 1000.2f);
System.out.println();
System.out.printf("$%,.2f", 1000.2f);
System.out.println();

```

We are combining multiple formatters here, but we are also prefixing the formatter with a dollar sign. You can add your formatter codes in any string literal when you use the `printf()` statement.

This example outputs the following:

```

1000.2
$1000.20
$1,000.20

```

## Spacing and Alignment Formatters

Often for text, you want to align or create a fixed amount of space for certain values or strings. You can use a string formatter to do exactly that by using the character `s` in your formatter code.

Check out this example.

**Listing 12-4.** Spacing and alignment formatters

```

System.out.println("Hello!");
System.out.printf("|%20s|", "Hello!");
System.out.println();
System.out.printf("|%-20s|", "Hello!");
System.out.println();
System.out.printf("|%-20s|", "Hello, this is a long sentence.!");
System.out.println();
System.out.printf("|%-20.20s|", "Hello, this is a long sentence.!");
System.out.println();

```

The output for this is

Hello!

```
|           Hello!|
|Hello!         |
|Hello, this is a long sentence.!!|
|Hello, this is a lon|
```

The pipes (the vertical lines) are added to make it clearer on what is going on here.

When you use a string formatter, the whole number value represents the minimum amount of characters the phrase is allowed to take up on the screen. So %20s will take up 20 characters of space on the screen.

If you add a minus sign to it, it will adjust the alignment, so it is on the other side. In this case, left aligned.

If you have a long string, it will print it out in its entirety, unless you add a decimal portion to the formatter. In this case, it will cut off any text that is longer than the maximum string length allowed by the formatter.

## Multiple Items in Formatters

You can add as many codes in your formatter string as you want; you just need to make sure you have a matching value to place inside of it. Each of these values is separated by commas.

**Listing 12-5.** Multiple items in a formatter

```
System.out.printf("%10s $%.2f\n","Apples",1.4f);
System.out.printf("%10s $%.2f\n","Brownies",0.8f);
```

The output for this one is

```
Apples $1.40
Brownies $0.80
```

Again, you can insert as many formatting codes as you want, but each value needs to be separated by a comma after it.

**Listing 12-6.** Sprint 12 code

```
public class Main {  
    public static void main(String[] args) {  
        /* String Formatting */  
  
        // Display a decimal number  
        System.out.println(2.3f);  
        System.out.printf("%.2f",2.3f);  
        System.out.println();  
        System.out.printf("%.4f",2.3f);  
        System.out.println();  
        /* Output  
        2.3  
        2.30  
        2.3000  
        */  
  
        // Display a thousands separator  
        System.out.println(1000000);  
        System.out.printf("%,d",1000000);  
        System.out.println();  
        /* Output  
        1000000  
        1,000,000  
        */  
  
        // Display a price  
        System.out.println(1000.2f);  
        System.out.printf("$%.2f", 1000.2f);  
        System.out.println();  
        System.out.printf("$%,.2f", 1000.2f);  
        System.out.println();  
    }  
}
```

## SPRINT 12 OUTPUT FORMATTING

```
/* Output:
1000.2
$1000.20
$1,000.20
*/

// Display a string
System.out.println("Hello!");
System.out.printf("|%-20s|", "Hello!");
System.out.println();
System.out.printf("|%-20s|", "Hello!");
System.out.println();
System.out.printf("|%-20s|", "Hello, this is a long sentence.!");
System.out.println();
System.out.printf("|%-20.20s|", "Hello, this is a long sentence.!");
System.out.println();
/* Output:
Hello!
|                Hello!|
|Hello!                |
|Hello, this is a long sentence.!!|
|Hello, this is a lon|
*/

// Display multiple items in a single string
System.out.printf("%10s $%.2f\n", "Apples", 1.4f);
System.out.printf("%10s $%.2f\n", "Brownies", 0.8f);
/* Output
    Apples $1.40
    Brownies $0.80
*/
}
}
```