

Format Specifiers: Formatted Element Output

This lesson explores the formatted element output part of the format specifiers.

WE'LL COVER THE FOLLOWING



- Formatted element output
 - Element format with strings
 - Element format for associative arrays

Formatted element output

Format specifiers between `%(and %)` are applied to every element of a container (e.g. an array or a range):

```
import std.stdio;

void main() {

    auto numbers = [ 1, 2, 3, 4 ];
    writeln("%(%s%", numbers);

}
```



The format string above consists of three parts:

- `%(` : start of element format
- `%s` : format for each element
- `%)` : end of element format

Each element is printed with the `%s` format and the elements appear one after the other:

The regular characters within the element format (% and %) are repeated for each element. For example, the `{%s}` specifier would print each element between curly brackets separated by commas:

```
import std.stdio;

void main() {

    auto numbers = [ 1, 2, 3, 4 ];
    writeln("%{%s},%", numbers);

}
```



However, regular characters to the right of the format specifier are considered to be element delimiters and are printed only between elements, not after the last one:

```
{1},{2},{3},{4 ← '}' and ',' are not printed after the last element
```

`%|` is used for specifying the characters that should be printed even for the last element. Characters that are to the right of `%|` are considered to be the delimiters and are not printed for the last element. Conversely, characters to the left of `%|` are printed even for the last element.

For example, the following format specifier would print the closing curly bracket after the last element but not the comma:

```
import std.stdio;

void main() {

    auto numbers = [ 1, 2, 3, 4 ];
    writeln("%{%s}%|,%", numbers);

}
```



Unlike previously, strings that are elements of an array get displayed within double quotes:

```
import std.stdio;

void main() {

    auto vegetables = [ "spinach", "asparagus", "artichoke" ];
    writeln("%(s, %)", vegetables);

}
```



When the double quotes are not desired, the element format must be started with `%-(` instead of `%(:`

```
import std.stdio;

void main() {

    auto vegetables = [ "spinach", "asparagus", "artichoke" ];
    writeln("%-(s, %)", vegetables);

}
```



The same applies to characters as well. `%('` prints them within single quotes:

```
import std.stdio;

void main() {

    writeln("%('s%)", "hello");

}
```



`%-(` prints them without quotes:

```
import std.stdio;

void main() {

    writeln("%-(s%)", "hello");

}
```



```
writeln( "%-(%s%)", hello );  
}
```



Element format for associative arrays

There must be two format specifiers for associative arrays: one for the keys and one for the values. For example, the following `%s (%s)` specifier would print first the key and then the value in parentheses:

```
import std.stdio;  
  
void main() {  
  
    auto spelled = [ 1 : "one", 10 : "ten", 100 : "hundred" ];  
    writeln("%-(%s (%s)%|, %)", spelled);  
}
```



Also note that, being specified to the right of `%|`, the comma is not printed for the last element:

```
1 (one), 100 (hundred), 10 (ten)
```

Next, we will see the positional parameters and format in format specifiers.