

011. 형식지정자를 사용하는 출력

1. 형식지정자 분류

1. 표준 형식지정자 (많이 사용)

- N - Number
- D - Decimal
- C - Currency
- F - Fixed Point
- E - Scientific

```
1 Console.WriteLine("{0:N2}", 1234.5678); // 출력: 1,234.57
2 Console.WriteLine("{0:D8}", 1234); // 출력: 00001234
3 Console.WriteLine("{0:F3}", 1234.56); // 출력: 1234.560
```

2. 커스텀 형식지정자

기호를 이용하여 쉽게 포맷을 지정할 수 있다.

- \# : Digit placeholder (0 이 앞에 붙지 않음)
- 0 : Zero placeholder (0 이 앞에 붙음)
- . : 소숫점 (Decimal point)
- , : 천 자리 (Thousands operator)
- ; : 섹션 구분 기호 (Section separator)
 - 섹션 구분 기호 (Section separator)는 아주 독특한 기능을 한다. 숫자를 표시할 때 양수, 음수, 0의 값을 세미콜론으로 구분하여 제각기 다른 포맷으로 출력할 수 있다. 예를 들어 회계에서는 음수 123 을 -123이 아니고, (123)으로 표시한다. 이럴 때 " { #,##0;(#,##0);zero } " 포맷을 쓰면 음수는 괄호 안에 숫자로, 0 은 zero로 출력한다.

2. 형식지정자를 사용하는 출력

형식지정자는 Console.WriteLine(), string.Format(), ToString() 에서 똑같이 사용할 수 있다.

3. 코드

```
1 using System;
2
3 namespace Application
4 {
5     public class A011_FormatSpecifier
6     {
7         static void Main(string[] args)
8         {
9             Console.WriteLine("{0:N2}", 1234.5678); // 출력: 1,234.57
10            Console.WriteLine("{0:D8}", 1234); // 출력: 00001234
```

```

11 Console.WriteLine("{0:F3}", 1234.56); // 출력: 1234.560
12 Console.WriteLine("{0,8}", 1234); // 출력: ____1234
13 Console.WriteLine("{0,-8}", 1234); // 출력: 1234_____
14
15 string s;
16 s = string.Format("{0:N2}", 1234.5678);
17 Console.WriteLine(s);
18 s = string.Format("{0:D8}", 1234);
19 Console.WriteLine(s);
20 s = string.Format("{0:F3}", 1234.56);
21 Console.WriteLine(s);
22
23 Console.WriteLine(1234.5678.ToString("N2"));
24 Console.WriteLine(1234.ToString("D8"));
25 Console.WriteLine(1234.56.ToString("F3"));
26
27 Console.WriteLine("{0:#.##}", 1234.5678);
28 Console.WriteLine("{0:0,0.00}", 1234.5678);
29 Console.WriteLine("{0:#,#.##}", 1234.5678);
30 Console.WriteLine("{0:000000.00}", 1234.5678);
31
32 Console.WriteLine("{0:#,#.##;(##.##);zero}", 1234.567);
33 Console.WriteLine("{0:#,#.##;(##.##);zero}", -1234.567);
34 Console.WriteLine("{0:#,#.##;(##.##);zero}", 0);
35 }
36 }
37 }

```

● Output

Terminal – Practice

```

1,234.57
00001234
1234.560
 1234
1234
1,234.57
00001234
1234.560
1,234.57
00001234
1234.560
1,234.57
1,234.57
1,234.57
001234.57
1,234.57
(1,234.57)
zero

```