

Java String (String, printf, format)

1. String String

“**String (Escape Sequence)** String String String String (String , , String) String String .
JavaString \ String .

1.1 String String

String	String	String String
<code>\n</code>	String	String String
<code>\t</code>	String	String (String 4~8String)
<code>\"</code>	String String	<code>"String String: \"String\""</code>
<code>\'</code>	String String	<code>'I\'m fine'</code>
<code>\\</code>	String String	<code>"C:\\Users\\Dain"</code>
<code>\r</code>	String String	String String String (String String)
<code>\b</code>	String	String String String (String String)
<code>\f</code>	String String	String String (String String)

1.2 String String

```
System.out.println("String\tString");  
System.out.println("String\t25");  
System.out.println("String String: \"String\"");  
System.out.println("C:\\Program Files\\Java");
```

이름 :

```
이름   이름
이름   25
이름   이름: "이름이름"
C:\Program Files\Java
```

2. ??? ?? ?? ??

Java이름 이름 이름 이름 이름 이름 System.out.printf() String.format()이름 이름 .

2.1 ?? ??

```
System.out.printf("이름", i1, i2...);
String result = String.format("이름", i1, i2...);
```

2.2 ?? ?? ??

이름 이름	이름	이름 이름
%s	이름	"이름: %s" → 이름 이름
%d	이름 (10이름)	"이름: %d"
%f	이름 (이름 이름)	"이름: %.2f"
%c	이름	"이름: %c"
%n	이름 (OS이름 이름)	"이름 이름"
%%	이름 (%) 이름	"100%% 이름"

2.3 ???/?? ??

이름	이름
%5d	이름 5이름 , 이름 이름
%-5d	이름 5이름 , 이름 이름
%05d	이름 0이름 이름 (이름 5이름)

<code>%.2f</code>	格式符 占位符 分隔符
<code>%6.2f</code>	格式符 宽度 精度 分隔符 占位符

2.4 变量 变量

```
String name = "小明";
int age = 25;
double score = 93.756;

System.out.printf("姓名: %s, 年龄: %d\n", name, age);
System.out.printf("分数: %.2f\n", score);
System.out.printf("总分: %d%%\n", 100);
```

输出结果：

```
姓名: 小明, 年龄: 25
分数: 93.76
总分: 100%
```

3. 转义 转义

转义符	转义符	转义符
换行符 制表符	<code>\n</code> , <code>\t</code> , <code>\\</code>	转义符 转义符 转义符
格式符 分隔符 (printf)	<code>%d</code> , <code>%.2f</code> , <code>%s</code>	转义符 转义符 转义符

4. 转义 转义 转义 (ANSI 转义)

Java 转义符 转义符 转义符 转义符 转义符 , 转义符 **ANSI** 转义符 转义符 转义符 转义符 转义符 转义符 .

```
public class ColorExample {
    public static final String RED = "\u001B[31m";
    public static final String RESET = "\u001B[0m";
```

```
public static void main(String[] args) {  
    System.out.println(RED + "  重置." + RESET);  
}  
}
```

	ANSI
	\u001B[31m
	\u001B[32m
	\u001B[33m
	\u001B[34m
	\u001B[0m

5. ?

Java java.time.LocalDateTime DateTimeFormatter /

5.1 DateTimeFormatter

```
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
  
LocalDateTime now = LocalDateTime.now();  
DateTimeFormatter fmt = DateTimeFormatter.ofPattern("yyyy-MM-dd HH:mm:ss");  
System.out.println("  : " + now.format(fmt));
```

5.2

yyyy		2025
MM		05
dd		22
HH	(24)	14

hh	HH (12時間)	02
mm		07
ss		59
a	AM / PM	AM / PM
E	曜日	Wed

5.3 日付フォーマッタ

```
DateTimeFormatter customFmt = DateTimeFormatter.ofPattern("yyyy-MM-dd (E) a hh:mm");
String formatted = now.format(customFmt);
System.out.println("日付フォーマッタ: " + formatted);
```

6. ログ

Javaには、`java.util.logging.Logger`というクラスがある。

6.1 Logger の使い方

```
import java.util.logging.*;

Logger logger = Logger.getLogger("MyLogger");

logger.info("Info ログ");
logger.warning("Warning ログ");
logger.severe("Severe ログ");
```

6.2 ログ出力のフォーマット

```
String name = "太郎";
int age = 25;
logger.info(String.format("名前: %s, 年齢: %d", name, age));
```

6.3 ?? ?? ??

??	??
SEVERE	??? ??
WARNING	??
INFO	?? ??
CONFIG	?? ?? ??
FINE	??? ??? ??

6.4 ?? ?? ???????

- ?? ?? ?? (?? /??)
 - ?? ??? ?? (?? , ?? ?? , ??? ?? ?)
 - logging.properties ??? ?? ?? ??
-
-