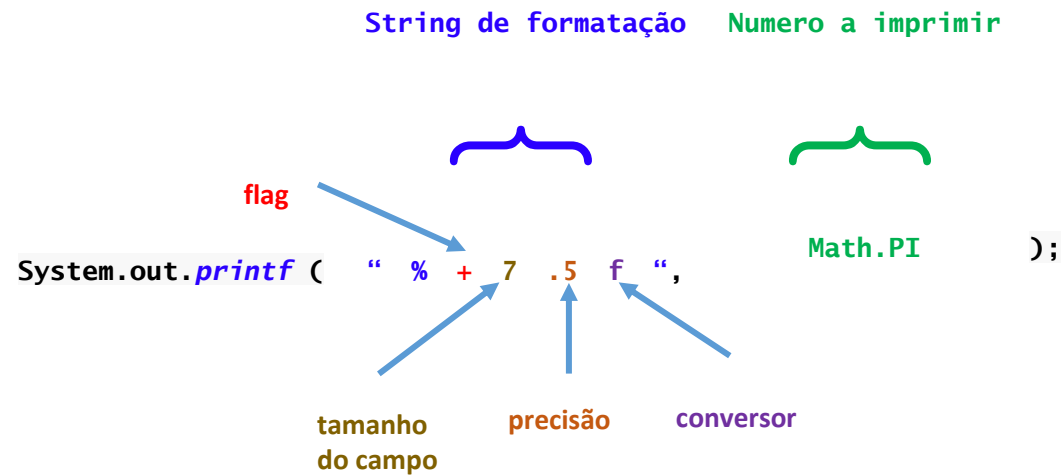


# JAVA - Formatação de saída Printf()

String de formatação = %[flags][width][.precision]conversion-character

[parâmetros entre colchetes são opcionais]



## Flags

- justificar à esquerda (o padrão é justificar à direita)
- + adiciona um sinal de mais ( + ) ou menos ( - ) para um valor numérico
- 0 força os valores numéricos a serem preenchidos com zeros (o padrão é preenchimento em branco)
- , separador de agrupamento de vírgulas (para números > 1000)
- espaço exibirá um sinal de menos se o número for negativo ou um espaço se for positivo

## JAVA - Formatação de saída Printf()

Caracteres de conversão		
d	inteiro decimal	[byte, short, int, long]
f	número de ponto flutuante	[float, double]
c	caractere	C maiúsculo colocará a letra em maiúscula
s	String	S maiúsculo colocará todas as letras na string em maiúsculas
h	hashcode	Um hashcode é como um endereço. Isso é útil para imprimir uma referência
n	newline	Caractere de nova linha específico da plataforma - use %n em vez de \n para maior compatibilidade
e	exponencial	notação científica de float ou double
b	booleano	valor booleano verdadeiro ou falso

```
System.out.printf("%+7.5f", Math.PI) OU System.out.println(String.format("%+7.5f", Math.PI))
```

Saída: +3,14159