

Tout savoir sur les spécificateurs de format en `printf` (%b, %s, etc.)

## Introduction

Dans le shell Bash (et en C également), la commande `printf` utilise des *spécificateurs de format*. Ceux-ci permettent de contrôler la manière dont les valeurs sont interprétées et affichées. Les plus courants sont %s, %d, %f, %x, %o, et, dans le cas de Bash, un très utile : %b.

Nous allons détailler chacun de ces spécificateurs, avec explications et exemples.

### %s : Chaîne de caractères

- Affiche la valeur comme une **chaîne brute**.
- Si la chaîne contient des séquences comme `\n` ou `\t`, elles *ne sont pas interprétées*.

**Exemple :**

```
1 printf "%s\n" "Bonjour\nMonde"
```

**Sortie :**

Bonjour\nMonde

Ici, les caractères `\n` restent littéraux.

### %b : Chaîne avec interprétation des échappements

- Identique à %s, mais interprète les séquences d'échappement du style C.
- Séquences courantes :
  - `\n` : retour à la ligne
  - `\t` : tabulation
  - `\033` ou `\e` : caractère ESC (utile pour les couleurs ANSI)
  - `\\` : un antislash littéral

**Exemple :**

```
1 printf "%b\n" "Bonjour\nMonde"
```

**Sortie :**

Bonjour  
Monde

## Comparaison directe

```

1 printf "%s\n" "Texte\nAvec\nSauts"
2 # Affiche : Texte\nAvec\nSauts
3
4 printf "%b\n" "Texte\nAvec\nSauts"
5 # Affiche :
6 # Texte
7 # Avec
8 # Sauts

```

## Autres spécificateurs utiles

- %d : entier décimal
- %f : flottant
- %x : entier en hexadécimal
- %o : entier en octal
- %c : caractère unique (correspondant au code ASCII)

### Exemple numérique :

```

1 printf "Dcimal : %d, Hexa : %x, Octal : %o\n" 65 65 65

```

### Sortie :

Décimal : 65, Hexa : 41, Octal : 101

## Largeur et alignement

- %10s : chaîne alignée à droite dans un champ de 10 caractères.
- %-10s : chaîne alignée à gauche.
- %05d : entier avec remplissage de zéros sur 5 chiffres.

### Exemple :

```

1 printf "|%10s|\n" "abc"
2 printf "|%-10s|\n" "abc"
3 printf "|%05d|\n" 42

```

### Sortie :

```

|          abc|
|abc         |
|00042|

```

## Applications pratiques de %b

- Affichage coloré (séquences ANSI) :

```

1 RED=$'\033[1;31m'
2 NC=$'\033[0m'
3 printf "%b\n" "${RED}Texte rouge${NC}"

```

- Multi-lignes intégrées :

```

1 printf "%b" "Ligne1\nLigne2\n"

```

## Résumé

- **%s** : affiche une chaîne littérale.
- **%b** : affiche une chaîne en interprétant les séquences d'échappement.
- Autres spécificateurs (**%d**, **%f**, **%x**, etc.) permettent un contrôle fin sur la présentation des nombres et chaînes.
- On peut combiner les options (largeur, alignement, remplissage).