

C# Formatage de chaînes de caractères

C# comme beaucoup d'autres langages de programmation utilise des caractères de format spéciaux lors de la construction d'une chaîne. Ces caractères affectent la manière dont les données sont présentées sur l'écran.

La commande C# pour le formatage est `String.Format`.

Cette méthode prend comme premier paramètre une chaîne de texte avec une liste de paramètres de type de données. Cependant, les pistes de format sont encore un peu difficiles à retenir, d'où ce petit guide sur le formatage de la chaîne C#.

Les caractères de format sont contenus dans les accolades et sont numérotés séquentiellement.

```
String.Format("C'est le nombre : {0}", 45);
```

Cela affichera "C'est le nombre : 45", en remplaçant le {0} par le premier paramètre de données. Le deuxième paramètre de données est remplacé par {1}, le troisième pour {2} et ainsi de suite.

Quelle que soit le type de la donnée à formater, `String.Format` se charge de la conversion en chaîne (appelle de la méthode `ToString()` pour chaque paramètre passé).

Formatage des valeurs numériques

Caractère	Description	Usage	Exemple de sortie
c	Devise	{0:c}	£12,345
d	Décimal	{0:d}	12345
e	Scientifique	{0:e}	1.234500e+004
f	Nombre à virgule fixe	{0:f}	12345
g	Général	{0:g}	12345
n	Séparateur de milliers	{0:n}	12,345
x	Hexadécimal (int seulement)	{0:x4}	3039

Formats généraux

Vous pouvez également utiliser des chaînes de format personnalisées, telles que les espaces réservés et caractères de début ou de fin de chaîne. Dans ces exemples, la valeur formatée est `12345.12`.

Caractère	Description	Usage	Exemple de sortie
0	Position des zéro	{0:00.0000}	12345.1200
#	Position des digits	{0:(#).##}	(12345).12
.	Séparateur décimal	{0:0.0}	12345.12
,	Séparateur de millier	{0:0,0}	12,345
%	Pourcentage	{0:0%}	1234512%
e	Notation scientifique	{0:00e+0}	12e+3

Formats de dates et heures

Les formats de dates et d'heures dépendent des paramètres régionaux (culture), de sorte que la sortie peut être différente.

Caractère	Description	Usage	Exemple de sortie
d	Date courte	{0:d}	08/12/2007
D	Date longue	{0:D}	08 December 2007
t	Heure (court)	{0:t}	15:27
T	Heure (long)	{0:T}	15:27:40
f	Date et heure (long)	{0:f}	08 December 2007 15:27
F	Date et heure (long)	{0:F}	08 December 2007 15:27:40
g	Date et heure (court)	{0:g}	08/12/2007 15:27
G	Date et heure (court)	{0:G}	08/12/2007 15:27:40
M	Date (court)	{0:M}	08 December
u	Universel	{0:u}	2007-12-08 15:27:40
U	Universel complète	{0:U}	08 December 2007 15:27:40
Y	Format mois année	{0:Y}	December 2007

Formats conditionnels

Vous pouvez également définir des conditions pour les valeurs numériques en utilisant le point-virgule. La syntaxe est `{0: positif; négatif; zéro}`.

Positif, Négatif ou Nul

```
// Sortie £123
string.Format("{0:&pound;#,##0.00; (£#,##0.00);Zero}",
123);

// Sortie £(123)
string.Format("{0:&pound;#,##0.00; (£#,##0.00);Zero}",
-123);

// Sortie Zero
string.Format("{0:&pound;#,##0.00; (£#,##0.00);Zero}", 0);
```

Oui/Non

```
// Sortie : Oui
string.Format("{0:Oui;;Non}", 1);

// Output Non
string.Format("{0:Oui;;Non}", 0);
```