**Formattazione composta**

La funzionalità di formattazione composta di .NET Framework consente di fornire un elenco di valori e una stringa di origine costituita da testo fisso alternato a segnaposto indicizzati e di ottenere quindi facilmente una stringa di risultati costituita dal testo fisso originale intervallato da valori formattati. Utilizzare la formattazione composta con metodi come String.Format, con cui viene restituita una stringa formattata, e Console.WriteLine, con cui la stringa di risultati viene visualizzata nella console, oppure con un'implementazione di TextWriter.WriteLine, con cui la stringa di risultati viene scritta in un file o in flusso.

Ciascun segnaposto indicizzato o elemento di formato corrisponde a un elemento in un elenco di valori. La funzionalità di formattazione composta restituisce una nuova stringa di risultati in cui ciascun elemento di formato incorporato nella stringa di origine è stato sostituito con il valore formattato corrispondente.

La stringa di origine è costituita da zero o più sequenze di testo fisso interrotte da uno o più elementi di formato. Il testo fisso può includere qualsiasi contenuto desiderato.

Si consideri come esempio l’utilizzo di **String.Format**:

**Metodo String.Format (String, Object)**

Consente di sostituire l'elemento di formato presente in un oggetto **String** specificato con l'equivalente in forma di testo del valore di un'istanza dell'oggetto **Object** specificata.

returnValue = String.Format(format, arg0)

Note

In questo metodo viene utilizzata la funzionalità di formattazione composta di .NET Framework per convertire il valore di un oggetto nella relativa rappresentazione testo e incorporare tale rappresentazione in una stringa. .NET Framework fornisce supporto di formattazione completo, descritto più dettagliatamente negli argomenti relativi alla formattazione riportati di seguito.

* Per ulteriori informazioni sulla funzionalità di formattazione composta supportata da metodi come Format e AppendFormat e da alcuni overload di WriteLine, vedere Formattazione composta.
* Per ulteriori informazioni sugli identificatori di formati numerici, vedere Stringhe di formato numerico standard e Stringhe di formato numerico personalizzato.
* Per ulteriori informazioni sugli identificatori di formati di data e ora, vedere Stringhe di formato DateTime standard e Stringhe di formato DateTime personalizzato.
* Per ulteriori informazioni sugli identificatori di formati di enumerazione, vedere Stringhe di formato di enumerazione.
* Per ulteriori informazioni sulla formattazione, vedere Formattazione dei tipi di dati e Cenni preliminari sulla formattazione.

Il parametro format è composto da zero o più esecuzioni di testo intervallate da zero o più segnaposti indicizzati, denominati elementi di formato, che corrispondono a un oggetto nell'elenco di parametri di questo metodo. Il processo di formattazione sostituisce ogni elemento di formato con la rappresentazione testo del valore dell'oggetto corrispondente.

La sintassi di un elemento di formato è **{*index*[,*alignment*][:*formatString*]}**, che specifica un indice obbligatorio, la lunghezza e l'allineamento facoltativi del testo formattato e una stringa facoltativa di caratteri identificatori di formato che controllano le modalità di formattazione del valore dell'oggetto corrispondente. I componenti di un elemento di formato sono:

* indice

Valore integer in base zero che indica l'elemento da formattare in un elenco di oggetti. Se l'oggetto specificato da *index* è riferimento null (**Nothing** in Visual Basic), l'elemento di formato verrà sostituito da una stringa vuota ("").

* allineamento

Valore integer facoltativo che indica la larghezza minima dell'area che conterrà il valore formattato. Se la lunghezza del valore formattato è minore di *alignment*, l'area dovrà essere riempita con spazi. Se *alignment* è negativo, il valore formattato sarà allineato a sinistra nell'area; se *alignment* è positivo, il valore formattato sarà allineato a destra. Se *alignment* non è specificato, la lunghezza dell'area corrisponderà alla lunghezza del valore formattato. La virgola è obbligatoria se *alignment* è specificato.

* formatString

Stringa facoltativa di identificatori di formato. I due punti sono obbligatori se *formatString* è specificato.

Sono obbligatorie anche le parentesi graffe iniziali e finali, "{" e "}". Per specificare un singolo carattere effettivo parentesi graffa in format, specificare due parentesi iniziali o finali, ossia "{{" o "}}".

Se il valore di format è {"Grazie per avere acquistato 0:####} copie di Microsoft®.NET (Core Reference)". e arg0 è un Int16 con valore 123, il valore restituito sarà:

"Grazie per aver acquistato 123 copie di Microsoft®.NET (Core Reference)."

Se il valore di format è "Il catalogo presenta {0,-8:G} voci.", arg0 è di tipo **Int16** con valore 42, e in questo esempio i segni di sottolineatura rappresentano gli spazi di riempimento, il valore restituito sarà:

"Il catalogo presenta 42\_\_\_\_\_\_ voci."

# Esempio:

int Numero = 18;

Console.WriteLine("Valore {0:###}", Numero);

Console.WriteLine(string.Format("Valore {0:000}", Numero));

Console.WriteLine(Numero.ToString("###"));

Console.WriteLine(string.Format("Valore {0,-10:000} Valore {1}", Numero, Numero));

Console.WriteLine(string.Format("Valore {0,10:000} Valore {1}", Numero, Numero));

Numero = 1234;

Console.WriteLine("Valore {0:###}", Numero);

Console.WriteLine("Valore {0:000}", Numero);

**Output:**

Valore 18

Valore 018

18

Valore 018 Valore 18

Valore 018 Valore 18

Valore 1234

Valore 1234

# Esempio

// This code example demonstrates the String.Format() method.

// Formatting for this example uses the "en-US" culture.

using System;

class Sample

{

enum Color {Yellow = 1, Blue, Green};

static DateTime thisDate = DateTime.Now;

public static void Main()

{

// Store the output of the String.Format method in a string.

string s = "";

Console.Clear();

// Format a negative integer or floating-point number in various ways.

Console.WriteLine("Standard Numeric Format Specifiers");

s = String.Format(

"(C) Currency: . . . . . . . . {0:C}\n" +

"(D) Decimal:. . . . . . . . . {0:D}\n" +

"(E) Scientific: . . . . . . . {1:E}\n" +

"(F) Fixed point:. . . . . . . {1:F}\n" +

"(G) General:. . . . . . . . . {0:G}\n" +

" (default):. . . . . . . . {0} (default = 'G')\n" +

"(N) Number: . . . . . . . . . {0:N}\n" +

"(P) Percent:. . . . . . . . . {1:P}\n" +

"(R) Round-trip: . . . . . . . {1:R}\n" +

"(X) Hexadecimal:. . . . . . . {0:X}\n",

-123, -123.45f);

Console.WriteLine(s);

// Format the current date in various ways.

Console.WriteLine("Standard DateTime Format Specifiers");

s = String.Format(

"(d) Short date: . . . . . . . {0:d}\n" +

"(D) Long date:. . . . . . . . {0:D}\n" +

"(t) Short time: . . . . . . . {0:t}\n" +

"(T) Long time:. . . . . . . . {0:T}\n" +

"(f) Full date/short time: . . {0:f}\n" +

"(F) Full date/long time:. . . {0:F}\n" +

"(g) General date/short time:. {0:g}\n" +

"(G) General date/long time: . {0:G}\n" +

" (default):. . . . . . . . {0} (default = 'G')\n" +

"(M) Month:. . . . . . . . . . {0:M}\n" +

"(R) RFC1123:. . . . . . . . . {0:R}\n" +

"(s) Sortable: . . . . . . . . {0:s}\n" +

"(u) Universal sortable: . . . {0:u} (invariant)\n" +

"(U) Universal sortable: . . . {0:U}\n" +

"(Y) Year: . . . . . . . . . . {0:Y}\n",

thisDate);

Console.WriteLine(s);

// Format a Color enumeration value in various ways.

Console.WriteLine("Standard Enumeration Format Specifiers");

s = String.Format(

"(G) General:. . . . . . . . . {0:G}\n" +

" (default):. . . . . . . . {0} (default = 'G')\n" +

"(F) Flags:. . . . . . . . . . {0:F} (flags or integer)\n" +

"(D) Decimal number: . . . . . {0:D}\n" +

"(X) Hexadecimal:. . . . . . . {0:X}\n",

Color.Green);

Console.WriteLine(s);

}

}

/\*

This code example produces the following results:

Standard Numeric Format Specifiers

(C) Currency: . . . . . . . . ($123.00)

(D) Decimal:. . . . . . . . . -123

(E) Scientific: . . . . . . . -1.234500E+002

(F) Fixed point:. . . . . . . -123.45

(G) General:. . . . . . . . . -123

(default):. . . . . . . . -123 (default = 'G')

(N) Number: . . . . . . . . . -123.00

(P) Percent:. . . . . . . . . -12,345.00 %

(R) Round-trip: . . . . . . . -123.45

(X) Hexadecimal:. . . . . . . FFFFFF85

Standard DateTime Format Specifiers

(d) Short date: . . . . . . . 6/26/2004

(D) Long date:. . . . . . . . Saturday, June 26, 2004

(t) Short time: . . . . . . . 8:11 PM

(T) Long time:. . . . . . . . 8:11:04 PM

(f) Full date/short time: . . Saturday, June 26, 2004 8:11 PM

(F) Full date/long time:. . . Saturday, June 26, 2004 8:11:04 PM

(g) General date/short time:. 6/26/2004 8:11 PM

(G) General date/long time: . 6/26/2004 8:11:04 PM

(default):. . . . . . . . 6/26/2004 8:11:04 PM (default = 'G')

(M) Month:. . . . . . . . . . June 26

(R) RFC1123:. . . . . . . . . Sat, 26 Jun 2004 20:11:04 GMT

(s) Sortable: . . . . . . . . 2004-06-26T20:11:04

(u) Universal sortable: . . . 2004-06-26 20:11:04Z (invariant)

(U) Universal sortable: . . . Sunday, June 27, 2004 3:11:04 AM

(Y) Year: . . . . . . . . . . June, 2004

Standard Enumeration Format Specifiers

(G) General:. . . . . . . . . Green

(default):. . . . . . . . Green (default = 'G')

(F) Flags:. . . . . . . . . . Green (flags or integer)

(D) Decimal number: . . . . . 3

(X) Hexadecimal:. . . . . . . 00000003

\*/

**Tabella della formattazione dei risultati numerici (Riferimenti per C#)**

Il formato viene specificato tramite le stringhe di formato. Nella tabella che segue sono contenute le stringhe di formato standard supportate. La stringa di formato presenta la forma **Axx**,, dove A è l'identificatore di formato e xx è l'identificatore di precisione. L**'identificatore di formato** controlla il tipo di formattazione applicato al valore numerico, mentre l**'identificatore di precisione** controlla il numero di cifre significative o decimali dell'output formattato.

**Carattere** **Descrizione** **Esempi** **Output**

C o c Valuta Console.Write("{0:C}", 2.5); $2.50

Console.Write("{0:C}", -2,5); ($2.50)

D o d Decimale Console.Write("{0:D5}", 25); 00025

E o e Scientifico Console.Write("{0:E}", 250000); 2.500000E+005

F o f A virgola fissa Console.Write("{0:F2}", 25); 25.00

Console.Write("{0:F0}", 25); 25

G o g Generale Console.Write("{0:G}", 2.5); 2.5

N o n Numero Console.Write("{0:N}", 2500000); 2,500,000.00

X o x Esadecimale Console.Write("{0:X}", 250); FA

Console.Write("{0:X}", 0xffff); FFFF

**Esempi di output di stringhe di formato numerico standard**

Nelle tabelle riportate di seguito sono contenuti alcuni esempi di output prodotti dal metodo **ToString** insieme agli identificatori di formato numerico standard e a tipi di dati, valori e lingue specifici.

### Valuta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| C | en-US | **Double** | 12345.6789 | $12,345.68 |
| C | de-DE | **Double** | 12345.678 | 12.345,68 € |

### Decimale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| D | en-US | **Int32** | 12345 | 12345 |
| D8 | en-US | **Int32** | 12345 | 00012345 |

### Scientifico (esponenziale)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| E | en-US | **Double** | 12345.6789 | 1.234568E+004 |
| E10 | en-US | **Double** | 12345.6789 | 1.2345678900E+004 |
| E | fr-FR | **Double** | 12345.6789 | 1,234568E+004 |
| e4 | en-US | **Double** | 12345.6789 | 1.2346e+004 |

### A virgola fissa

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| F | en-US | **Double** | 12345.6789 | 12345.68 |
| F | es-ES | **Double** | 12345.6789 | 12345,68 |
| F0 | en-US | **Double** | 12345.6789 | 123456 |
| F6 | en-US | **Double** | 12345.6789 | 12345.678900 |

**Generale**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| G | en-US | **Double** | 12345.6789 | 12345.6789 |
| G7 | en-US | **Double** | 12345.6789 | 12345.68 |
| G | en-US | **Double** | 0.0000023 | 2.3E-6 |
| G | en-US | **Double** | 0.0023 | 0.0023 |
| G2 | en-US | **Double** | 1234 | 1.2E3 |
| G | en-US | **Double** | Pi greco | 3.14159265358979 |

### Numerico

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| N | en-US | **Double** | 12345.6789 | 12,345.68 |
| N | sv-SE | **Double** | 12345.6789 | 12 345,68 |
| N4 | en-US | **Double** | 123456789 | 123,456,789.0000 |

### Percentuale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| P | en-US | **Double** | .126 | 12.60 % |

### Esadecimale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Stringa di formato** | **Lingua** | **Tipo di dati** | **Valore** | **Output** |
| x | en-US | **Int32** | 0x2c45e | 2c45e |
| X | en-US | **Int32** | 0x2c45e | 2C45E |
| X8 | en-US | **Int32** | 0x2c45e | 0002C45E |
| x | en-US | **Int32** | 123456789 | 75bcd15 |

**Stringhe di formato numerico personalizzato**

Una stringa di formato numerico personalizzato creata dall'utente e costituita da uno o più identificatori di formato numerico personalizzato definisce la formattazione dei dati numerici. Viene considerata stringa di formato numerico personalizzato qualsiasi stringa che non rientra nella categoria di stringa di formato numerico standard. Nella tabella riportata di seguito vengono descritti gli identificatori di formato numerico personalizzato

|  |  |  |
| --- | --- | --- |
| **Identificatore di formato** | **Nome** | **Descrizione** |
| 0 | Segnaposto per zero | Se il valore da formattare dispone di una cifra nella posizione in cui nella stringa di formato si trova uno "0", tale cifra verrà copiata nella stringa di risultati. La posizione dell'ultimo "0" a sinistra prima del separatore decimale e dell'ultimo "0" a destra dopo il separatore decimale determina l'intervallo di cifre sempre presenti nella stringa di risultati.  L'identificatore "00" determina l'arrotondamento del valore alla cifra più vicina prima del decimale, in cui viene sempre utilizzato l'arrotondamento a un valore diverso da zero. La formattazione di 34,5 con "00", ad esempio, restituisce come risultato il valore 35. |
| # | Segnaposto per cifre | Se il valore da formattare dispone di una cifra nella posizione in cui nella stringa di formato si trova uno "#", tale cifra verrà copiata nella stringa di risultati. In caso contrario, nella stringa di risultati non verrà memorizzato alcun valore in tale posizione.  Si noti che questo identificatore non visualizza mai il carattere "0" se non si tratta di una cifra significativa, anche se "0" è l'unica cifra della stringa. Il carattere "0" viene visualizzato solo se si tratta di una cifra significativa nel numero da visualizzare.  La stringa di formato "##" determina l'arrotondamento del valore alla cifra più vicina prima del decimale, in cui viene sempre utilizzato l'arrotondamento a un valore diverso da zero. La formattazione di 34,5 con "##", ad esempio, restituisce come risultato il valore 35. |
| . | Separatore decimale | Il primo carattere "," della stringa di formato determina la posizione del separatore decimale nel valore formattato. Eventuali ulteriori caratteri "," vengono ignorati. |
| , | Separatore delle migliaia | Identificatore del separatore delle migliaia: se viene specificato uno o più caratteri ',' tra due segnaposti per cifre (0 o #) che formattano le cifre integrali di un numero, verrà inserito un carattere di separazione di gruppi tra ogni gruppo di numeri nella parte integrale dell'output. |

**Esempi di output di stringhe di formato numerico personalizzato**

Nella tabella riportata di seguito viene illustrato l'output creato applicando alcune stringhe di formato numerico personalizzato a specifici tipi di dati e valori. L'output è stato generato utilizzando il metodo **ToString** e la lingua Inglese (Stati Uniti) (en-US).

|  |  |  |  |
| --- | --- | --- | --- |
| **Stringa di formato** | **Tipo di dati** | **Valore** | **Output** |
| ##### | **Double** | 123 | 123 |
| 00000 | **Double** | 123 | 00123 |
| (###) ### - #### | **Double** | 1234567890 | (123) 456 – 7890 |
| #.## | **Double** | 1.2 | 1.2 |
| 0.00 | **Double** | 1.2 | 1.20 |
| 00.00 | **Double** | 1.2 | 01.20 |
| #,# | **Double** | 1234567890 | 1,234,567,890 |
| #,, | **Double** | 1234567890 | 1235 |
| #,,, | **Double** | 1234567890 | 1 |
| #,##0,, | **Double** | 1234567890 | 1,235 |
| #0.##% | **Double** | 0.086 | 8.6% |
| 0.###E+0 | **Double** | 86000 | 8,6E+4 |
| 0.###E+000 | **Double** | 86000 | 8,6E+004 |
| 0.###E-000 | **Double** | 86000 | 8,6E004 |
| [##-##-##] | **Double** | 123456 | [12-34-56] |
| ##;(##) | **Double** | 1234 | 1234 |
| ##;(##) | **Double** | -1234 | (1234) |

**Stringhe di formato DateTime standard**

Nella tabella riportata di seguito vengono descritti gli identificatori di formato **DateTime** standard.

|  |  |  |
| --- | --- | --- |
| **Identificatore di formato** | **Nome** | **Descrizione** |
| d | Schema di data breve | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà ShortDatePattern corrente.  La stringa di formato personalizzato per la lingua inglese ad esempio è "MM/dd/yyyy". |
| D | Schema di data esteso | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà LongDatePattern corrente. La stringa di formato personalizzato per la lingua inglese ad esempio è "dddd, dd MMMM yyyy". |
| f | Schema di data/ora completo (ora breve) | Rappresenta una combinazione dello schema di data esteso (D) e dello schema di ora breve (t), separati da uno spazio. |
| F | Schema di data/ora completo (ora estesa) | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà FullDateTimePattern corrente. La stringa di formato personalizzato per la lingua inglese ad esempio è "dddd, dd MMMM yyyy HH:mm:ss". |
| g | Schema di data/ora generale (ora breve) | Rappresenta una combinazione dello schema di data breve (d) e dello schema di ora breve (t), separati da uno spazio. |
| G | Schema di data/ora generale (ora estesa) | Rappresenta una combinazione dello schema di data breve (d) e dello schema di ora esteso (T), separati da uno spazio. |
| M o m | Schema mese giorno | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà MonthDayPattern corrente. La stringa di formato personalizzato per la lingua inglese ad esempio è "MMMM dd". |
| s | Schema di data/ora ordinabile; conforme a ISO 8601 | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà SortableDateTimePattern corrente. Lo schema è uno standard definito e la proprietà è di sola lettura. Sarà pertanto sempre lo stesso, indipendentemente dalla lingua utilizzata o dal provider di formato fornito.  La stringa di formato personalizzato è "yyyy'-'MM'-'dd'T'HH':'mm':'ss". |
| t | Schema di ora breve | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà ShortTimePattern corrente.  La stringa di formato personalizzato per la lingua inglese ad esempio è "HH:mm". |
| T | Schema di ora esteso | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà LongTimePattern corrente. La stringa di formato personalizzato per la lingua inglese ad esempio è "HH:mm:ss". |
| u | Schema di data/ora ordinabile universale | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà UniversalSortableDateTimePattern corrente. Lo schema è uno standard definito e la proprietà è di sola lettura. Sarà pertanto sempre lo stesso, indipendentemente dalla lingua utilizzata o dal provider di formato fornito.  La stringa di formato personalizzato è "yyyy'-'MM'-'dd HH':'mm':'ss'Z".  Non viene eseguita alcuna conversione del fuso orario con la formattazione di data e ora. L'applicazione dovrà pertanto convertire la data e l'ora locali in formato UTC (Coordinated Universal Time) per utilizzare questo identificatore di formato. |
| Y o y | Schema anno mese | Rappresenta una stringa di formato **DateTime** personalizzato definita dalla proprietà YearMonthPattern corrente. La stringa di formato personalizzato per la lingua inglese ad esempio è "yyyy MMMM". |

**Esempi di output di stringhe di formato DateTime standard**

Nella tabella riportata di seguito viene illustrato l'output creato applicando alcune stringhe di formato **DateTime** standard a una data e a un'ora specifiche. L'output è stato ottenuto mediante il metodo **ToString**.

Nella colonna Stringa di formato viene indicato l'identificatore di formato, nella colonna Lingua viene riportata la lingua associata al thread corrente e nella colonna Output viene mostrato il risultato della formattazione.

I diversi valori di lingua consentono di illustrare l'impatto della modifica della lingua corrente. La lingua può essere cambiata utilizzando le impostazioni di **Opzioni internazionali e della lingua** nel Pannello di controllo. La modifica della lingua non influisce sull'output prodotto dai formati "r" e "s".

### Schema di data breve

**Stringa di formato Lingua corrente Output**

d en-US 4/10/2001

d en-NZ 10/04/2001

d de-DE 10.04.2001

### Schema di data esteso

**Stringa di formato Lingua corrente Output**

D en-US Tuesday, April 10, 2001

### Schema di ora esteso

**Stringa di formato Lingua corrente Output**

T en-US 3:51:24 PM

T es-ES 15:51:24

### Schema di data/ora completo (ora breve)

**Stringa di formato Lingua corrente Output**

f en-US Tuesday, April 10, 2001 3:51 PM

f fr-FR mardi 10 avril 2001 15:51

### Schema di data/ora ordinabile (ISO 8601)

**Stringa di formato Lingua corrente Output**

s en-US 2001-04-10T15:51:24

s pt-BR 2001-04-10T15:51:24

### Schema di data/ora ordinabile universale

**Stringa di formato Lingua corrente Output**

u en-US 2001-04-10 15:51:24Z

u sv-FI 2001-04-10 15:51:24Z

### Schema mese giorno

**Stringa di formato Lingua corrente Output**

m en-US April 10

m ms-MY 10 April

### Schema anno mese

**Stringa di formato Lingua corrente Output**

y en-US April, 2001

y af-ZA April 2001

**Stringhe di formato DateTime personalizzato**

Una stringa di formato **DateTime** personalizzato è costituita da uno o più identificatori di formato **DateTime** personalizzato e definisce la rappresentazione di testo di un oggetto DateTime derivante da un'operazione di formattazione. Qualsiasi stringa che non è una stringa di formato DateTime standard viene interpretata come stringa di formato **DateTime** personalizzato.

# Identificatori di formato DateTime personalizzato

Nella tabella riportata di seguito vengono descritti gli identificatori di formato **DateTime** personalizzato.

|  |  |
| --- | --- |
| **Identificatore di formato** | **Descrizione** |
| d | Rappresenta il giorno del mese come numero compreso tra 1 e 31. I giorni costituiti da una singola cifra vengono formattati senza zero iniziale. |
| dd | Rappresenta il giorno del mese come numero compreso tra 01 e 31. I giorni costituiti da una singola cifra vengono formattati con uno zero iniziale. |
| ddd | Rappresenta il nome abbreviato del giorno della settimana corrente. |
| dddd (più un numero qualsiasi di identificatori "d" aggiuntivi) | Rappresenta il nome completo del giorno della settimana corrente. |
| f | Rappresenta la cifra più significativa della frazione di secondi.  Se viene utilizzato da solo, senza altri identificatori di formato, l'identificatore di formato "f" viene interpretato come identificatore di formato **DateTime** standard "f" (schema di data/ora completo). |
| ff | Rappresenta le due cifre più significative della frazione di secondi. |
| fff | Rappresenta le tre cifre più significative della frazione di secondi. |
| ffff | Rappresenta le quattro cifre più significative della frazione di secondi. |
| fffff | Rappresenta le cinque cifre più significative della frazione di secondi. |
| ffffff | Rappresenta le sei cifre più significative della frazione di secondi. |
| fffffff | Rappresenta le sette cifre più significative della frazione di secondi. |
| F | Rappresenta la cifra più significativa della frazione di secondi. Se la cifra è zero, non viene prodotta alcuna visualizzazione. |
| FF | Rappresenta le due cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con due zeri, non vengono visualizzati. |
| FFF | Rappresenta le tre cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con tre zeri, non vengono visualizzati. |
| FFFF | Rappresenta le quattro cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con quattro zeri, non vengono visualizzati. |
| FFFFF | Rappresenta le cinque cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con cinque zeri, non vengono visualizzati. |
| FFFFFF | Rappresenta le sei cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con sei zeri, non vengono visualizzati. |
| FFFFFFF | Rappresenta le sette cifre più significative della frazione di secondi. Gli zeri finali, o le cifre con sette zeri, non vengono visualizzati. |
| h | Rappresenta l'ora come numero compreso tra 1 e 12, ovvero l'ora rappresentata nel formato delle 12 ore in base al quale il conteggio riparte da mezzanotte o da mezzogiorno. Di conseguenza, una particolare ora dopo mezzanotte è indistinguibile dalla stessa ora dopo mezzogiorno. L'ora non viene arrotondata e se è costituita da una singola cifra viene formattata senza zero iniziale. Per un valore 5:43 ad esempio questo identificatore di formato visualizza "5". |
| hh, hh (più un numero qualsiasi di identificatori "h" aggiuntivi) | Rappresenta l'ora come numero compreso tra 01 e 12, ovvero l'ora rappresentata nel formato delle 12 ore in base al quale il conteggio riparte da mezzanotte o da mezzogiorno. Di conseguenza, una particolare ora dopo mezzanotte è indistinguibile dalla stessa ora dopo mezzogiorno. L'ora non viene arrotondata e se è costituita da una singola cifra viene formattata con uno zero iniziale. Per un valore 5:43 ad esempio questo identificatore di formato visualizza "05". |
| H | Rappresenta l'ora come numero compreso tra 0 e 23, ovvero l'ora rappresentata nel formato delle 24 ore in base zero in cui il conteggio parte da mezzanotte. Un'ora costituita da una singola cifra viene formattata senza zero iniziale. |
| HH, HH (più un numero qualsiasi di identificatori "H" aggiuntivi) | Rappresenta l'ora come numero compreso tra 00 e 23, ovvero l'ora rappresentata nel formato delle 24 ore in base zero in cui il conteggio parte da mezzanotte. Un'ora costituita da una singola cifra viene formattata con uno zero iniziale. |
| m | Rappresenta il minuto come numero compreso tra 0 e 59. Il minuto rappresenta minuti interi trascorsi dall'ultima ora. Un minuto costituito da una singola cifra viene formattato senza zero iniziale. |
| mm, mm (più un numero qualsiasi di identificatori "m" aggiuntivi) | Rappresenta il minuto come numero compreso tra 0 e 59. Il minuto rappresenta minuti interi trascorsi dall'ultima ora. Un minuto costituito da una singola cifra viene formattato con uno zero iniziale. |
| M | Rappresenta il mese come numero compreso tra 1 e 12. Un mese costituito da una singola cifra viene formattato senza zero iniziale. |
| MM | Rappresenta il mese come numero compreso tra 01 e 12. Un mese costituito da una singola cifra viene formattato con uno zero iniziale. |
| MMM | Rappresenta il nome abbreviato del mese corrente. |
| MMMM | Rappresenta il nome completo del mese corrente. |
| s | Rappresenta i secondi come numero compreso tra 0 e 59. Il secondo rappresenta secondi interi trascorsi dall'ultimo minuto. Un secondo costituito da una singola cifra viene formattato senza zero iniziale.. |
| ss, ss (più un numero qualsiasi di identificatori "s" aggiuntivi) | Rappresenta i secondi come numero compreso tra 00 e 59. Il secondo rappresenta secondi interi trascorsi dall'ultimo minuto. Un secondo costituito da una singola cifra viene formattato con uno zero iniziale. |
| y | Rappresenta l'anno con al massimo un numero a due cifre. Se l'anno ha più di due cifre, nel risultato vengono visualizzate solo le due cifre di ordine inferiore. Se l'anno ha meno di due cifre, il numero viene formattato senza zero iniziale. |
| yy | Rappresenta l'anno come numero a due cifre. Se l'anno ha più di due cifre, nel risultato vengono visualizzate solo le due cifre di ordine inferiore. Se l'anno ha meno di due cifre, al numero vengono anteposti tanti zeri quanto sono necessari per ottenere due cifre. |
| yyy | Rappresenta l'anno come numero a tre cifre. Se l'anno ha più di tre cifre, nel risultato vengono visualizzate solo le tre cifre di ordine inferiore. Se l'anno ha meno di tre cifre, al numero vengono anteposti tanti zeri quanti sono necessari per ottenere tre cifre. |
| yyyy | Rappresenta l'anno come numero a quattro cifre. Se l'anno ha più di quattro cifre, nel risultato vengono visualizzate solo le quattro cifre di ordine inferiore. Se l'anno ha meno di quattro cifre, al numero vengono anteposti tanti zeri quanto sono necessari per ottenere quattro cifre. |
| yyyyy (più un numero qualsiasi di identificatori "y" aggiuntivi) | Rappresenta l'anno come numero a cinque cifre. Se l'anno ha più di cinque cifre, nel risultato vengono visualizzate solo le cinque cifre di ordine inferiore. Se l'anno ha meno di cinque cifre, al numero vengono anteposti tanti zeri quanto sono necessari per ottenere cinque cifre. Se esistono ulteriori identificatori "y", al numero vengono anteposti tanti zeri quanti sono necessari per ottenere il numero di identificatori "y". |

**Esempi di output di stringhe di formato DateTime personalizzato**

Nella tabella riportata di seguito viene illustrato l'output creato applicando alcune stringhe di formato **DateTime** personalizzato a una data e a un'ora specifiche.

I diversi valori di lingua e di fuso orario consentono di illustrare l'impatto della modifica della lingua corrente. La lingua può essere modificata utilizzando l'icona **Data e ora** del Pannello di controllo.

|  |  |  |  |
| --- | --- | --- | --- |
| **Stringa di formato** | **Lingua corrente** | **Fuso orario** | **Output** |
| d, M | en-US | GMT | 12, 4 |
| d, M | es-MX | GMT | 12, 4 |
| d MMMM | en-US | GMT | 12 April |
| d MMMM | es-MX | GMT | 12 Abril |
| dddd MMMM yy gg | en-US | GMT | Thursday April 01 A.D. |
| dddd MMMM yy gg | es-MX | GMT | Jueves Abril 01 DC |
| h , m: s | en-US | GMT | 6 , 13: 12 |
| hh,mm:ss | en-US | GMT | 06,13:12 |
| HH-mm-ss-tt | en-US | GMT | 06-13-12-AM |