#### Zahlen (Wert = 1000000)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Symbol | Typ | Aufruf | Ergebnis | | | |
| c | Währung (currency) | string.Format("{0:c}",Wert) | | 1.000.000,00 € | |
| d | Dezimalzahl (decimal) | string.Format("{0:d}",Wert) | | | 1000000 | |
| e | Wissenschaftlich (scientific) | string.Format("{0:e}",Wert) | | | 1,000000e+006 | |
| f | Festkommazahl (fixed point) | string.Format("{0:f}",Wert) | | | 1000000,00 | |
| g | Generisch (general) | string.Format("{0:g}",Wert) | | | 1000000 | |
| n | Tausender trennzeichen | string.Format("{0:n}",Wert) | | | 1.000.000,00 | |
| x | Hexadezimal | string.Format("{0:x4}",Wert) | | | f4240 | |

#### Zahlen individuell formatieren (Wert = 1000000)

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Typ | Aufruf | Ergebnis |
| 0 | 0-Platzhalter | string.Format("{0:00.0000}",Wert) | 1000000,0000 |
| # | Zahl-Platzhalter | string.Format("{0:(#).##}",Wert) | (1000000) |
| . | Dezimalpunkt | string.Format("{0:0.0}",Wert) | 1000000,0 |
| , | Tausender trennzeichen | string.Format("{0:0,0}",Wert) | 1.000.000 |
| ,. | Ganzzahliges Vielfaches von 1.000 | string.Format("{0:0,.}",Wert) | 1000 |
| % | Prozentwert | string.Format("{0:0%}",Wert) | 100000000% |
| e | Exponenten-Platzhalter | string.Format("{0:00e+0}",Wert) | 10e+5 |

#### Datumswerte formatieren (Wert = aktuelles Datum = System.DateTime.Now)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Symbol | Typ | Aufruf | | Ergebnis | |
| d | kurzes Datumsformat | | string.Format("{0:d}",Wert) | | 26.02.2016 | |
| D | langes Datumsformat | | string.Format("{0:D}",Wert) | | Freitag, 26. Februar 2016 | |
| t | kurzes Zeitformat | | string.Format("{0:t}",Wert) | | 09:56 | |
| T | langes Zeitformat | | string.Format("{0:T}",Wert) | | 09:56:48 | |
| f | Datum & Uhrzeit komplett (kurz) | | string.Format("{0:f}",Wert) | | Freitag, 26. Februar 2016 09:56 | |
| F | Datum & Uhrzeit komplett (lang) | | string.Format("{0:F}",Wert) | | Freitag, 26. Februar 2016 09:56:48 | |
| g | Standard-Datum (kurz) | | string.Format("{0:g}",Wert) | | 26.02.2016 09:56 | |
| G | Standard-Datum (lang) | | string.Format("{0:G}",Wert) | | 26.02.2016 09:56:48 | |
| M | Tag des Monats | | string.Format("{0:M}",Wert) | | 26 Februar | |
| r | RFC1123 Datumsformat | | string.Format("{0:r}",Wert) | | Fri, 26 Feb 2016 09:56:48 GMT | |
| s | sortierbares Datumsformat | | string.Format("{0:s}",Wert) | | 2016-02-26T09:56:48 | |
| u | universell sortierbares Datumsformat | | string.Format("{0:u}",Wert) | | 2016-02-26 09:56:48Z | |
| U | universell sortierbares GMT-Datumsformat | | string.Format("{0:U}",Wert) | | Freitag, 26. Februar 2016 08:56:48 | |
| Y | Jahr/Monats-Muster | | string.Format("{0:Y}",Wert) | | Februar 2016 | |

#### Datumswerte individuell formatieren (Wert = aktuelles Datum = System.DateTime.Now)

|  |  |  |  |
| --- | --- | --- | --- |
| Symbol | Typ | Aufruf | Ergebnis |
| dd | Tag | string.Format("{0:dd}",Wert) | 26 |
| ddd | Tagname (Kürzel) | string.Format("{0:ddd}",Wert) | Fr |
| dddd | Tagname (ausgeschrieben) | string.Format("{0:dddd}",Wert) | Freitag |
| gg | Ära | string.Format("{0:gg}",Wert) | n. Chr. |
| hh | Stunde 2stellig | string.Format("{0:hh}",Wert) | 09 |
| HH | Stunde 2stellig (24-Stunden) | string.Format("{0:HH}",Wert) | 09 |
| mm | Minute | string.Format("{0:mm}",Wert) | 56 |
| MM | Monat | string.Format("{0:MM}",Wert) | 02 |
| MMM | Monatsname (Kürzel) | string.Format("{0:MMM}",Wert) | Feb |
| MMMM | Monatsname (ausgeschrieben) | string.Format("{0:MMMM}",Wert) | Februar |
| ss | Sekunde | string.Format("{0:ss}",Wert) | 48 |
| tt | AM oder PM (nur englisch) | string.Format("{0:tt}",Wert) |  |
| yy | Jahr 2stellig | string.Format("{0:yy}",Wert) | 16 |
| yyyy | Jahr 4stellig | string.Format("{0:yyyy}",Wert) | 2016 |
| zz | Zeitzone (kurz) | string.Format("{0:zz}",Wert) | +01 |
| zzz | Zeitzone (lang) | string.Format("{0:zzz}",Wert) | +01:00 |