

# Hur formaterar jag numeriska värden?

Följande kod skriver ut värdet 483 i konsolfönstret:

```
Console.WriteLine("Värdet: {0}.", 483); // Skriver ut talet.
Console.WriteLine("Värdet: {0:c}.", 483); // Formatera det som valuta.
```

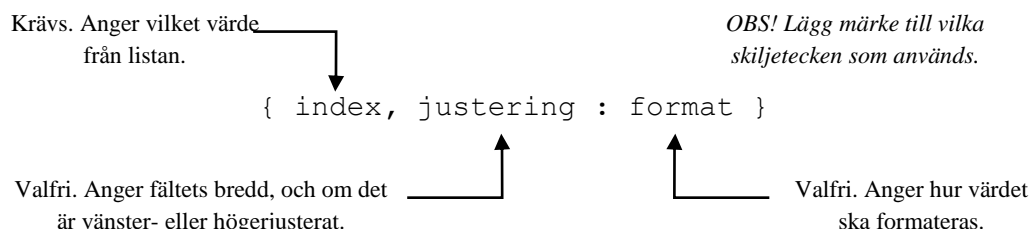
De två satserna producerar följande:



Figur 1. I konsolfönstret presenteras talet 483 olika beroende på hur det formateras

Skillnaden mellan de två satserna är att den andra använder en formatspecificerare som i detta fall automatiskt lägger till information om valuta.

En formatspecificerare består av tre fält mellan två klammerparenteser: ett index, justering och formatet.



## Justeringsfältet

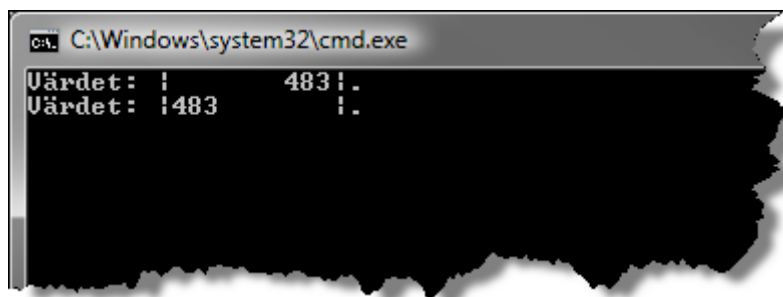
Värdet i justeringsfältet anger den *minsta bredd* för fältet räknat i antal tecken. Justeringsfältet kan innehålla ett positivt eller negativt heltal:

- ✓ Heltalet anger minsta antalet tecken som används.
- ✓ Ett positivt heltal ger högerjustering inom fältet; negativt heltal ger vänsterjustering.

Satserna som följer får illustrera hur det fungerar:

```
int myValue = 483;
Console.WriteLine("Värdet: |{0, 10}|.", myValue); // Högerjusterat.
Console.WriteLine("Värdet: |{0, -10}|.", myValue); // Vänsterjusterat.
```

För att det ska bli enklare att se effekten av justeringen så presenteras värdet som myValue har mellan två vertikallinjer (|).



Figur 2. Effekten av höger- och vänsterjustering

Skulle det vara så att värdet som ska presenteras behöver fler tecken än vad som anges så ignoreras justeringsfältet och värdet presenteras med så många tecken som krävs.

## Formatfältet

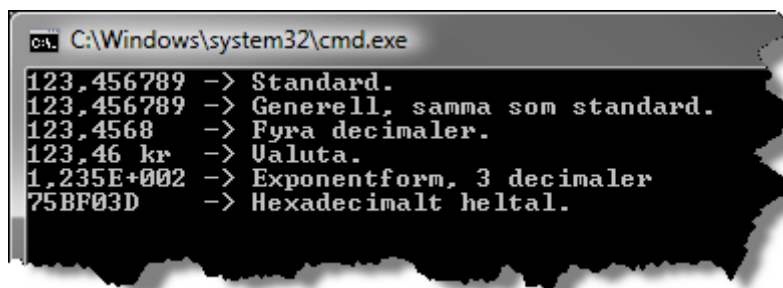
Formatfältet anger på vilket sätt som det numeriska värdet ska presenteras. Ska det presenteras som en valuta, med ett visst antal decimaler eller som procent?

Formatfältet består av två delar:

- ✓ Formatspecificerare är en bokstav, en av nio som kan användas.
- ✓ Precisionsspecificerare, som är valfri, består av en eller två siffror. Siffrornas betydelse beror av vilken formatspecificerare som används.

Satserna som följer visar hur formatfältet påverkar hur ett värde presenteras.

```
double myValue = 123.456789;
Console.WriteLine("{0, -10} -> Standard.", myValue);
Console.WriteLine("{0, -10:G} -> Generell, samma som standard.", myValue);
Console.WriteLine("{0, -10:F4} -> Fyra decimaler.", myValue);
Console.WriteLine("{0, -10:C} -> Valuta.", myValue);
Console.WriteLine("{0, -10:E3} -> Exponentform, 3 decimaler", myValue);
Console.WriteLine("{0, -10:X} -> Hexadecimalt heltal.", 123465789);
```



Figur 3. Hur ett värde presenteras kan påverkas av formatfältet

Se onlinedokumentationen för mer information om olika sätt att formatera numeriska värden.

- ✓ *Composite Formatting*,  
[http://msdn2.microsoft.com/en-us/library/txafckwd\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/txafckwd(VS.80).aspx)
- ✓ *Standard Numeric Format Strings*,  
[http://msdn2.microsoft.com/en-us/library/dwhawy9k\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/dwhawy9k(VS.80).aspx)