

Att omvandla tal till text (strängar med siffror) kan göras på olika sätt. De två bästa sätten att konvertera värdet av `x` till en sträng är att använda en f-sträng eller att anropa metoden `format` i strängklassen.

f-strängar

Antag att vi har satt två variabler `n` och `x` till 12 respektive 2/3 dvs. ett värde av heltals- och ett av flyttalstyp.

Exempel på konvertering med dessa tal:

Kod	Värde
<code>f{n}</code>	'12'
<code>f n = {n}</code>	'n = 12'
<code>f{n} i kvadrat är {n*n}</code>	'12 i kvadrat är 144'
<code>f{x}</code>	'0.6666666666666666'
<code>f x i kvadrat är {x**2}</code>	'x i kvadrat är 0.4444444444444444'

I f-strängar kan man alltså skriva ett uttryck inom klammerparenteser. Dessa uttryck evalueras till ett värde som sedan konverteras till en sträng.

Det går att specificera detaljer som fältbredd och antal decimaler genom att skriva en *formatkod*.

Exempel med samma `n` och `x` som ovan men med formatkoder:

Kod	Värde	Kommentar
<code>f{n:6d}</code>	' 12'	Heltal, 6 positioner, högerjusterat.
<code>f{n:<6d}</code>	'12 '	Heltal, 6 positioner, vänsterjusterat.
<code>f{n:^6d}</code>	' 12 '	Heltal, 6 positioner, centrerat.
<code>f{x:.3f}</code>	'0.667'	Flyttal, 3 decimaler.
<code>f{x:10.4f}</code>	' 0.6667'	Flyttal, 4 decimaler, högerjusterat i 10 positioner.
<code>f{x:.5e}</code>	'6.66667e-01'	Flyttal på exponentform, 5 decimaler.

Således:

- Formatkoderna skrivs efter kolon inom klammrarna.
- Koden `d` anger heltal på decimal form.
- Koden `f` anger flyttal på decimal form.
- Koden `e` anger flyttal på exponentform.
- Fältets bredd kan anges efter bokstavskoden.
- Positionering i fältet kan specificeras med `<` och `^`. Högerjustering är default men kan också anges med `>`.
- För flyttal kan antal decimaler specificeras efter en punkt.

Metoden `format`

Samma exempel som med f-strängar ovan:

Kod	Värde
<code>'{}'.format(n)</code>	<code>'12'</code>
<code>'n = {}'.format(n)</code>	<code>'n = 12'</code>
<code>'{} i kvadrat är {}'.format(n, n*n)</code>	<code>'12 i kvadrat är 144'</code>
<code>'{}'.format(x)</code>	<code>'0.6666666666666666'</code>
<code>'{:6d}'.format(n)</code>	<code>' 12'</code>
<code>'{:<6d}'.format(n)</code>	<code>'12 '</code>
<code>'{: ^6d}'.format(n)</code>	<code>' 12 '</code>
<code>'{: .3f}'.format(x)</code>	<code>'0.667'</code>
<code>'{:10.4f}'.format(x)</code>	<code>' 0.6667'</code>
<code>'{: .5e}'.format(x)</code>	<code>'6.66667e-01'</code>

Här anger man alltså värdena som användas som parametrar till `format`-metoden. Även här kan man ange formatkoder efter kolontecknet i *platshållaren* `{}`.

Formatering med `%`-operatorn

Exempel:

Kod	Värde
<code>'n = %d, x = %5.2f' % (n, x)</code>	<code>'n = 12, x = 0.67'</code>

Här skrivs formatkoderna efter `%` inuti strängen. Tecknet `%` används också som operator mellan en sträng och en tupel.

Detta är ett föråldrat sätt att göra formatkonverteringar, men man bör i alla fall förstå vad det är när man ser det.