

# Programmering med Python 3 (version 3.7.2)

## Skriva ut data

OPERATION	RESULTAT
<code>print(1.5)</code>	1.5
<code>print(1.5 + 2)</code>	3.5
<code>print(1,5 + 2)</code>	1 7
<code>print("Svar: Sju")</code>	Svar: Sju
<code>print("Svar:" + "Sju")</code>	Svar:Sju
<code>print("Svar:" , "Sju")</code>	Svar: Sju
<code>print("Svar:", 1.5+2)</code>	Svar: 3.5

## Matematiska operationer

OPERATION	KOMMENTAR
<code>x + y</code>	Summan av x och y.
<code>x - y</code>	Differensen av x och y.
<code>x * y</code>	Produkten av x och y.
<code>x/y</code>	Kvoten av x och y.
<code>x//y</code>	Heltalsdelen vid divisionen x/y.
<code>x % y</code>	Resten vid divisionen x/y.
<code>x**y</code>	x upphöjt till y.
<code>abs(x)</code>	Absolutvärdet av x.
<code>round(x, n)</code>	Avrundar flyttalet x till n antal decimaler.

## Variabler

OPERATION	KOMMENTAR
<code>tal = 7</code>	Sparar talet 7 i variabeln tal.
<code>tal = 3 + 2</code>	Sparar resultatet 5 i variabeln tal.
<code>tal = tal + 2</code>	Variabeln tal får värdet tal + 2.
<code>namn = "Sara"</code>	Sparar texten (strängen) Sara i variabeln namn.

## Datatyper

<code>int()</code>	Heltal
<code>float()</code>	Decimtal (flyttal)
<code>complex(re, im)</code>	Komplext tal med realdelen re och imaginärdelen im.
<code>str()</code>	Sträng (text och symboler)

## Skriva in data

### OPERATION

```
namn = input("Vad heter du?")
heltal = int(input("Skriv ett heltal."))
decimal = float(input("Skriv ett tal
                    med decimaler."))
```

### KOMMENTAR

Inmatning sparas som en textsträng.  
Inmatning sparas som ett heltal.  
Inmatning sparas som ett tal med decimaler.

## Jämförelser, logiska operationer

### OPERATION

### KOMMENTAR

### EXEMPEL

<code>x == y</code>	Sant om x är lika med y.	<code>7 == 7</code> ger True.
<code>x != y</code>	Sant om x inte är lika med y.	
<code>x &gt; y</code>	Sant om x är större än y.	
<code>x &lt; y</code>	Sant om x är mindre än y.	<code>7 &lt; 3</code> ger False.
<code>x &gt;= y</code>	Sant om x är större än eller lika med y.	
<code>x &lt;= y</code>	Sant om x är mindre än eller lika med y.	
<code>x and y</code>	Sant om både x och y är sanna.	<code>7 &gt; 3 and 7 &lt; 3</code> ger False.
<code>x or y</code>	Sant om x och/eller y är sant.	<code>7 == 7 or 7 &lt; 3</code> och <code>7 == 7 or 3 == 3</code> ger True.
<code>not x</code>	Sant om x är falskt.	<code>not(7 == 7)</code> ger False.

## if-satser

### EXEMPEL

```
vinkel = int(input("Ange en vinkel i grader."))
if vinkel < 90:
    print("spetsig")
elif vinkel == 90:
    print("rät")
else:
    print("trubbig")
```

### KOMMENTAR

Frågar efter en vinkel i en triangel.  
Om vinkel är mindre än 90 grader, skrivs spetsig.  
Om vinkeln är 90 grader, skrivs rät.  
Annars skrivs trubbig.

## for-loopar

### EXEMPEL

### KOMMENTAR

```
for i in range(4):
    print(i)
```

for i in range(1, 4):

Loopen körs 4 gånger. Programmet skriver ut talen 0, 1, 2 och 3.

Loopen körs 4 – 1 gånger. Programmet skriver talen 1, 2 och 3.

## while-loopar

### EXEMPEL

```
x = 0
while x < 4:
    print(x)
    x = x + 1
```

### KOMMENTAR

Variabeln  $x$  sätts till 0.  
Så länge  $x$  är mindre än 4, utförs instruktionerna:  
 $x$  skrivs ut och  
 $x$  ökar med 1.

## Listor

### OPERATION

```
lista = []
lista = [2, 4, 6, 2, 6]
lista[0]
lista[2]
lista.append(8)
lista.index(4)
lista.extend(lista2)
lista.insert(2, 5)
lista.remove(2)
lista.pop(3)

lista.sort()
lista.reverse()
lista.count(2)
sum(lista)
len(lista)
max(lista)
min(lista)
```

### RESULTAT

Skapar den tomma listan lista.  
Skapar lista som innehåller i ordning talen 2, 4, 6, 2, 6.  
Ger det första elementet i lista (som har index 0).  
Ger det tredje elementet (6) i lista (som har index 2).  
Lägger till talet 8 sist i lista.  
Ger index för den första 4:an i lista.  
Lägger till lista2 till slutet av lista.  
Lägger in 5 i lista på platsen framför index 2.  
Tar bort den första förekomsten av 2 ur lista.  
Ger det element i lista som har index 3. Tar sedan bort det elementet från listan.  
Sorterar lista i storleksordning med minsta talet först.  
Elementen i lista läggs i omvänd ordning.  
Räknar antal förekomster av elementet 2 i lista.  
Summerar alla tal i lista.  
Ger antalet element i lista.  
Ger det största talet i lista.  
Ger det minsta talet i lista.

## Importera matematiska funktioner

```
from math import *
```

Importerar samtliga funktioner från modulen math.  
Dessa funktioner är endast definierade för reella tal.

```
sqr(x)
exp(x)
log(x)
log10(x)
log(x, y)
sin(x), cos(x), tan(x)

asin(x), acos(x), atan(x)
```

Kvadratroten ur  $x$ .  
Talet  $e$  upphöjt till  $x$ .  
Den naturliga logaritmen av  $x$ .  
10-logaritmen av  $x$ .  
 $y$ -logaritmen av  $x$ .  
De trigonometriska funktionerna. Argumentet anges i radianer.  
De inversa trigonometriska funktionerna. Resultatet är en vinkel i radianer.

## Importera matematiska funktioner (forts.)

<code>degrees(x)</code>	Omvandlar vinkeln $x$ från radianer till grader.
<code>radians(x)</code>	Omvandlar vinkeln $x$ från grader till radianer.
<code>pi</code>	Konstanten $\pi$ , $\pi = 3,141592\dots$
<code>e</code>	Konstanten $e$ , $e = 2,718281\dots$
<code>from random import *</code>	Importerar samtliga funktioner från modulen <i>random</i> .
<code>randint(a,b)</code>	Slumpar fram ett heltal $N$ i intervallet $a \leq N \leq b$ .
<code>uniform(a,b)</code>	Slumpar fram ett flyttal $R$ i intervallet $a \leq R \leq b$ .
<code>choice(lista)</code>	Slumpar fram ett element ur listan <i>lista</i> .
<code>from statistics import *</code>	Importerar samtliga funktioner från modulen <i>statistics</i> .
<code>median(lista)</code>	Beräknar medianen av talen i <i>lista</i> .
<code>mean(lista)</code>	Beräknar medelvärdet av talen i <i>lista</i> .
<code>mode(lista)</code>	Beräknar typvärdet bland elementen i <i>lista</i> .
<code>pstdev(lista)</code>	Beräknar standardavvikelsen där <i>lista</i> utgör hela populationen.
<code>stdev(lista)</code>	Beräknar standardavvikelsen där <i>lista</i> utgör ett stickprov av hela populationen.

## Komplexa tal

<code>from cmath import *</code>	Importerar samtliga funktioner från modulen <i>cmath</i> . Dessa funktioner är definierade för komplexa tal.
<code>z = complex(3,7)</code>	Sparar det komplexa talet $3 + 7j$ i variabeln <i>z</i> .
<code>z.conjugate()</code>	Det komplexa konjugatet ( $3 - 7j$ ) till <i>z</i> .
<code>z.real</code>	Realdelen (3) till <i>z</i> för det komplexa talet i variabeln <i>z</i> .
<code>z.imag</code>	Imaginärdelen (7) av komplexa talet i variabeln <i>z</i> .

\*I Python har den imaginära enheten beteckningen *j*.

## Definiera egna funktioner

### EXEMPEL

```
def cylinderns_area(r,h):
    return 3.14*r**2*h

print(cylinderns_area(2,3))

def fakultet(n):
    res = 1
    for i in range(n):
        res = res*(i+1)
    return res
```

### KOMMENTAR

Definierar en funktion `cylinderns_area`, som tar radien *r* och höjden *h* som argument. Beräknar och returnerar volymen av en cylinder med radien *r* och höjden *h*.

Anropar funktionen `cylinderns_area` och skriver ut det returnerade värdet 37.68.

Definierar funktionen `fakultet` med *n* som argument. Loopar *n* gånger  $i = 0, 1, 2, \dots, n - 1$ .

Returnerar värdet av produkten  $1 \cdot 2 \cdot 3 \cdot \dots \cdot n$ .