

---

## Allmänt om Mathematica

- Utvecklades av Wolfram Research (Stephen Wolfram) på 80-talet
- Programmet finns bl.a. till Windows, Mac OS X, Linux.
- Finns (åtminstone) installerat i ASA B121 (Stansen), i matematik institutionens datasal, samt på [tuxedo.abo.fi](http://tuxedo.abo.fi)
- Mest använd för symboliska beräkningar



## Köra *Mathematica* på [tuxedo.abo.fi](http://tuxedo.abo.fi)

- Man kan köra *Mathematica* på unixservern [tuxedo.abo.fi](http://tuxedo.abo.fi) i textläge.
- Inte speciellt användarvänligt, men kan lätt utföra enkla beräkningar.
- För att starta programmet loggar man in t.ex. med SSH på [tuxedo.abo.fi](http://tuxedo.abo.fi) och skriver kommandot "math".
- Avsluta med kommandot "Quit".



## Användargränssnittet

- Användargränssnittet är uppbyggt kring s.k. notebooks och flytande verktygspaletteer.
- Klarar sig helt utan verktygspaletteerna, men de underlättar i många fall.
  - I version 7 finns bland annat "Basic Math Assistant" som innehåller de flesta funktioner man behöver
- Kommandon skrivs in i ett inputfält i en notebook och evalueras med SHIFT + ENTER
- Vanliga aritmetiska uttryck kan användas.

```
2 + 5 ^ 2 - 4 Sqrt [ 9 ] + Sin [ 2 Pi / 3 ]
```

$$15 + \frac{\sqrt{3}}{2}$$

- Observera att inbyggda funktioner och konstanter skrivs med stor bokstav.
  - Funktioner: Sqrt[ ], Exp[ ], Cos[ ], ArcSin[ ].
  - Konstanter: E, I, Infinity, Pi
- Vanliga parenteser ( ) används för att gruppera uttryck.



## Hjälpfunktioner och manual

- Mathematica har en utmärkt inbyggd manual som hittas i "Help" menyn.
- För att få hjälp om ett specifikt kommando används `?Funktionsnamn`

```
? Sqrt
```

`Sqrt[z]` or  $\sqrt{z}$  gives the square root of  $z$ . [>>](#)

- Kan även trycka F1 då markören befinner sig på ett kommando.

◀ | ▶

## Uttryck och symboler i Mathematica

### ■ Numerisk evaluering (N)

```
N[Pi]
```

```
3.14159
```

```
N[Pi, 50]
```

```
3.1415926535897932384626433832795028841971693993751
```

```
E // N
```

```
2.71828
```

### ■ Ge variabler ett värde (=) och tömma variabler (.=)

```
a = 5 Pi - 2
```

```
-2 + 5 π
```

```
a^2
```

```
(-2 + 5 π)2
```

```
N[a, 10]
```

```
13.70796327
```

```
a = .  
a
```

```
a
```

### ■ Jämförelser ( ==, >, >=, <, <=, != )

```
2 == 1
```

```
False
```

```
2 > 1
```

```
True
```

### ■ Skriv ej ut resultatet (;)

```
N[Pi];
```

```
a = 2 + Exp[1];
```

```
a
```

```
2 + e
```

### ■ Listor (Räckor, Arrays) { }

```
A = {1, 5, 6, 3, 10}
```

```
{1, 5, 6, 3, 10}
```

```
A[[3]]
```

```
6
```

```
A[[2 ;; 4]]
```

```
{5, 6, 3}
```

## ■ Ersätta värden i uttryck (/ . eller ReplaceAll)

- För att ersätta variabler i uttryck med ett värde, en annan variabel eller ett uttryck används /. (snedstreck och punkt). Är en kortform för kommandot ReplaceAll

```
Clear[a, b, c];  
a = b + c
```

```
b + c
```

```
a /. b → 1
```

```
1 + c
```

```
ReplaceAll[a, {b → Cos[y], c → Sin[x]}]
```

```
Cos[y] + Sin[x]
```

- Detta ändrar inte värdet på det ursprungliga uttrycket:

```
a
```

```
b + c
```

- Pilen skrivs med ett streck och ett större än tecken (*Mathematica* ersätter det automatiskt med en pil) eller så använder man någon av verktygspaletterna.

## ■ Referera till tidigare erhållna resultat

```
2 + 3
```

```
5
```

$\% + 10$ 

15

 $\%25 + \%26$ 15 +  $\%26$ 

- Obs! Spara hellre svaret till en variabel, eftersom referenserna ändras vartefter nya uttryck beräknas.





## Funktioner

### ■ Funktioner defineras enligt

```
f[x_] := x^2;  
g[x_, y_] := x + y;
```

- Sedan kan de användas som vilken inbyggd funktion som helst.
- Observera `_`-tecknet efter variabeln.

```
f[2] + g[4, 3]
```

```
11
```

- För att kontrollera en funktions definition skriv `?Funktionsnamn`.

```
?f
```

```
Global`f
```

```
f[x_] := x2
```

### ■ Begränsa definitionsmängden

```
Clear[h]  
h[x_] := Sqrt[x] /; x ≥ 0  
?h
```

```
Global`h
```

```
h[x_] :=  $\sqrt{x}$  /; x ≥ 0
```

```
h[3]  
h[-1]
```

```
 $\sqrt{3}$ 
```

```
h[-1]
```

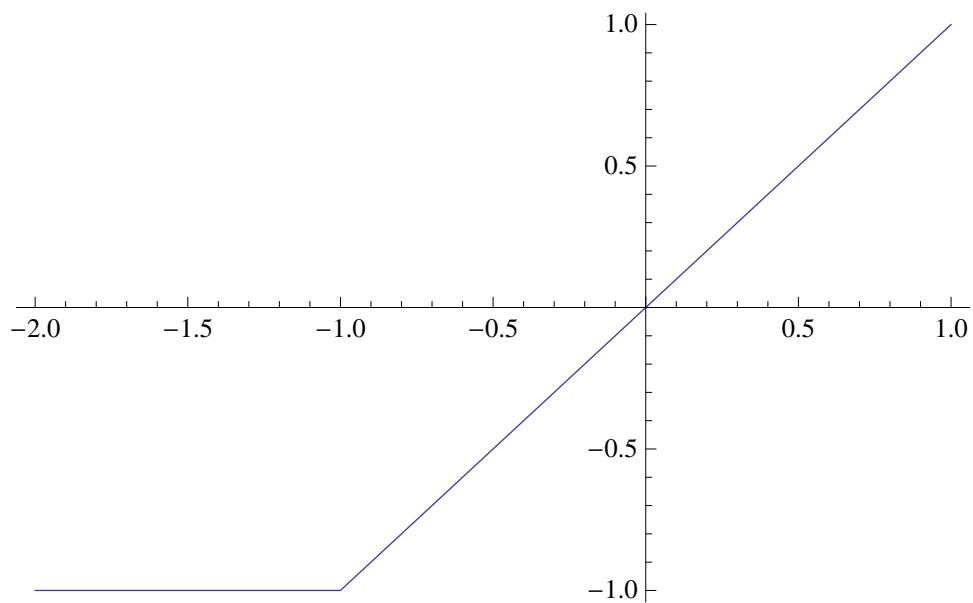
## ■ Styckevis definerade funktioner

### □ Alternativ 1

```
f[x_] := -1 /; x < -1  
f[x_] := x /; x ≥ -1  
f[-100]  
f[100]  
Plot[f[x], {x, -2, 1}]
```

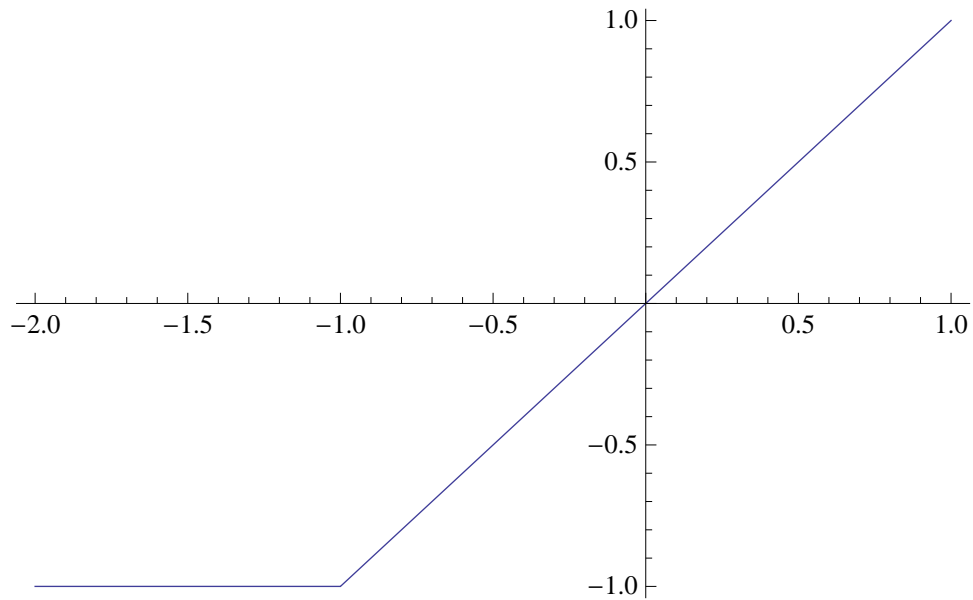
```
-1
```

```
100
```



### □ Alternativ 2

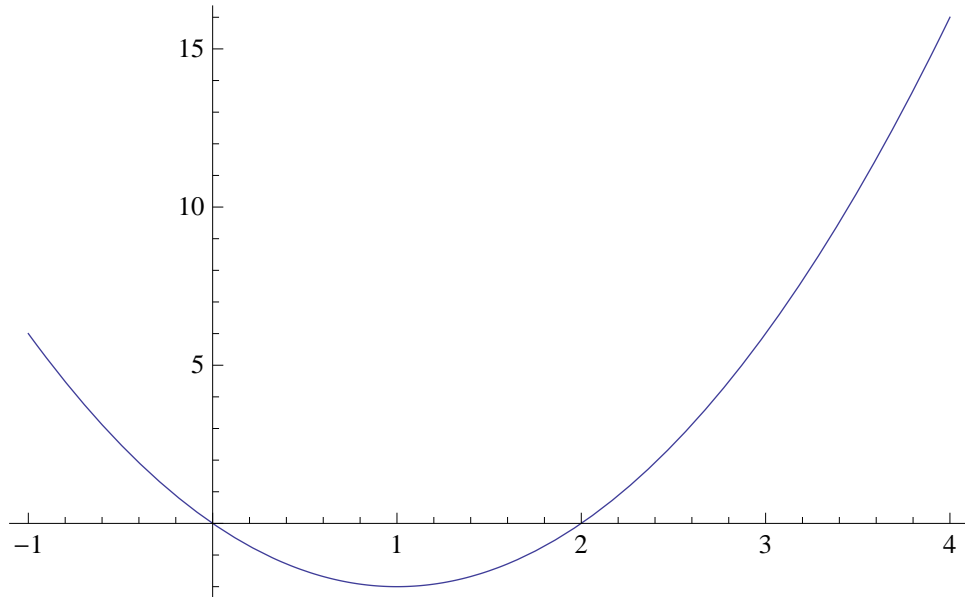
```
g[x_] := Piecewise[{{-1, x < -1}, {x, x ≥ -1}}]  
Plot[g[x], {x, -2, 1}]
```



### ■ Anropa funktioner med flera värden (/@ eller Map)

- För att anropa en funktion med flera givna värden kan man använda Map-funktionen eller dess kortform `/@`

```
Clear[f];  
f[x_] := 2 x^2 - 4 x;  
Plot[f[x], {x, -1, 4}]
```



```
f /@ {0, 1, 2, 3}  
Map[f, {0, 1, 2, 3}]
```

```
{0, -2, 0, 6}
```

```
{0, -2, 0, 6}
```

- Om man har en funktion av flera variabler, kan man använda `MapThread` istället:

```
g[x_, y_] := 2 x + y;  
MapThread[g, {{1, 2, 3}, {0, 1, 1}}]
```

```
{2, 5, 7}
```

## Aritmetik

- I verktygsfältet "Basic Math Assistant" -> "Basic Commands" -> "y==x" finns flera olika funktioner för att omforma uttryck.

$$p = (2 + x) (3 + y) (4 + z)^2$$

$$(2 + x) (3 + y) (4 + z)^2$$

$$p2 = \text{Expand}[p]$$

$$96 + 48 x + 32 y + 16 x y + 48 z + 24 x z + \\ 16 y z + 8 x y z + 6 z^2 + 3 x z^2 + 2 y z^2 + x y z^2$$

$$\text{Factor}[p2]$$

$$(2 + x) (3 + y) (4 + z)^2$$

$$\text{FullSimplify}[p2 + p2^2]$$

$$(2 + x) (3 + y) (4 + z)^2 \\ (97 + 2 y (4 + z)^2 + x (3 + y) (4 + z)^2 + 6 z (8 + z))$$



## Vektorer och matriser

### ■ Vektorer

- Mathematica tolkar vektorer som listor.

```
a = {1, 2, -1}; b = {2, -2, 3};
```

- Addition och subtraktion av vektorer fungerar som vanligt.

```
a + b
```

```
{3, 0, 2}
```

- Inre produkten fås som den ena produkten "punkt" den andra.

```
a.b
```

```
-5
```

### ■ Matriser

- Mathematica tolkar matriser som listor av vektorer, alltså dubbellistor.

```
A = {{1, 3}, {-4, 9}}; B = {{2, 3}, {-1, 1}};
```

```
MatrixForm[A]
```

```
MatrixForm[B]
```

```
( 1 3 )
( -4 9 )
```

```
( 2 3 )
( -1 1 )
```

- Addition och subtraktion av matriser fungerar som vanligt.

```
A + B // MatrixForm
```

$$\begin{pmatrix} 3 & 6 \\ -5 & 10 \end{pmatrix}$$

- Matrimultiplikation är den ena matrisen "punkt" den andra.

```
A.B // MatrixForm
```

$$\begin{pmatrix} -1 & 6 \\ -17 & -3 \end{pmatrix}$$

## ■ Matrisfunktioner

### □ Determinanten

```
Det[A]
```

21

### □ Transponatet

```
MatrixForm[Transpose[B]]
```

$$\begin{pmatrix} 2 & -1 \\ 3 & 1 \end{pmatrix}$$

### □ Matrisinvers

```
MatrixForm[Inverse[A]]
```

$$\begin{pmatrix} \frac{3}{7} & -\frac{1}{7} \\ \frac{4}{21} & \frac{1}{21} \end{pmatrix}$$

## □ Egenvärden och egenvektorer

```
Eigenvalues[A]  
Eigenvectors[A]  
Eigensystem[A]
```

```
{7, 3}
```

```
{{1, 2}, {3, 2}}
```

```
{{7, 3}, {{1, 2}, {3, 2}}}
```





## Integrering och differentiering

### ■ Differentiering

- Differentiering av en funktion kan göras genom att använda ' - tecknet.

```
f[x_] := Exp[-x^2];  
f'[x]
```

$$-2 e^{-x^2} x$$

```
f''[x]
```

$$-2 e^{-x^2} + 4 e^{-x^2} x^2$$

- Differentiering av en funktion kan även göras med hjälp av D- funktionen.

```
D[f[x], x]
```

$$-2 e^{-x^2} x$$

- Högre ordningens derivator fås på följande sätt

```
D[x^x, {x, 2}]
```

$$x^{-1+x} + x^x (1 + \text{Log}[x])^2$$

- **Exempel: Gör en funktion som ger ut n-te ordningens derivata av den naturliga logaritmfunktionen**

```
Clear[x, n];
DLog[n_] := D[Log[x], {x, n}];
DLog /@ {1, 2, 10}
```

$$\left\{ \frac{1}{x}, -\frac{1}{x^2}, -\frac{362880}{x^{10}} \right\}$$

## ■ Integrering

- Integrering kan göras genom att använda någon av paletterna "Basic Math Assistant" eller "Classroom Assistant"

$$\int_0^1 x^3 dx$$

$$\frac{1}{4}$$

$$\int \sin[x] dx$$

$$-\cos[x]$$

- Integrering kan även göras med hjälp av Integrate-funktionen.

```
Integrate[x^3, {x, 0, 1}]
```

$$\frac{1}{4}$$

```
Integrate[Sin[x], x]
```

$$-\cos[x]$$

- Ifall *Mathematica* inte klarar av att utföra en integration returneras

funktionen själv:

```
Integrate[Sin[x] / Log[x], x]
```

$$\int \frac{\sin[x]}{\log[x]} dx$$

- Numerisk integration utförs med `NIntegrate`-kommandot:

```
NIntegrate[Sin[x] / Log[x], {x, 2, 10}]
```

```
0.275074
```



## Summor och produkter

### ■ Summor

- Summor görs antingen med hjälp av paletterna eller så använder man `Sum`-kommandot.

$$\sum_{k=1}^{\infty} \frac{1}{k^2}$$

$$\frac{\pi^2}{6}$$

```
Sum[1/k^2, {k, 1, Infinity}]
```

$$\frac{\pi^2}{6}$$

- Ifall summan divergerar:

```
Sum[1/k, {k, 1, Infinity}]
```

Sum::div : Sum does not converge. >>

$$\sum_{k=1}^{\infty} \frac{1}{k}$$

### ■ Produkter

- Produkter fås på samma sätt som summor, men kommandot är `Product` istället för `Sum`

$$\prod_{i=1}^n i^2$$

$$(n!)^2$$

```
Product[1 - 1 / i^4, {i, 2, Infinity}]
```

$$\frac{\text{Sinh}[\pi]}{4 \pi}$$

- Ifall produkten ej konvergerar:

```
Product[1 + 1 / i, {i, 1, Infinity}]
```

Product::div: Product does not converge. >>

$$\prod_{i=1}^{\infty} \left(1 + \frac{1}{i}\right)$$



## Gränsvärden

- Gränsvärden bestäms med `Limit`-kommandot

```
Limit[Sin[x] / x, x → 0]
```

1

```
Limit[(1 + x/n)^n, n → Infinity]
```

$e^x$

```
Limit[1/x, x → 0, Direction → -1]
```

$\infty$

```
Limit[1/x, x → 0, Direction → +1]
```

$-\infty$

## Ekvationslösning

- Ekvationer med en eller flera obekanta variabler löses med `Solve`-kommandot.

```
Clear[a, b, c, x];
Solve[a x^2 + b x + c == 0, x]
```

$$\left\{ \left\{ x \rightarrow \frac{-b - \sqrt{b^2 - 4 a c}}{2 a} \right\}, \left\{ x \rightarrow \frac{-b + \sqrt{b^2 - 4 a c}}{2 a} \right\} \right\}$$

```
Solve[2 x^2 - 2 x - 10 == 0, x]
N[%]
```

$$\left\{ \left\{ x \rightarrow \frac{1}{2} (1 - \sqrt{21}) \right\}, \left\{ x \rightarrow \frac{1}{2} (1 + \sqrt{21}) \right\} \right\}$$

$$\left\{ \left\{ x \rightarrow -1.79129 \right\}, \left\{ x \rightarrow 2.79129 \right\} \right\}$$

- Resultatet ges som ersättningsuttryck (alltså med  $\rightarrow$ ). Vill man ha dem som punkter kan man göra på följande sätt

```
sol = Solve[2 x^2 - 2 x - 10 == 0, x]
x /. sol
```

$$\left\{ \left\{ x \rightarrow \frac{1}{2} (1 - \sqrt{21}) \right\}, \left\{ x \rightarrow \frac{1}{2} (1 + \sqrt{21}) \right\} \right\}$$

$$\left\{ \frac{1}{2} (1 - \sqrt{21}), \frac{1}{2} (1 + \sqrt{21}) \right\}$$

- För att lösa ekvationssystem används `&&` mellan de olika ekvationerna. Observera att även ett högerled måste finnas med, t.ex. `== 0`.

```
Solve[a x + y == 7 && b x - y == 1, {x, y}]
```

$$\left\{ \left\{ x \rightarrow \frac{8}{a+b}, y \rightarrow -\frac{a-7b}{a+b} \right\} \right\}$$

- **Exempel: Hitta skärningsspunkterna mellan en cirkel och en parabel**

```
pts = Solve[x^2 + y^2 == 1 && y - 2 x^2 + 3/2 == 0, {x, y}]
```

$$\left\{ \left\{ y \rightarrow \frac{1}{4} (-1 - \sqrt{5}), x \rightarrow -\frac{1}{2} \sqrt{\frac{1}{2} (5 - \sqrt{5})} \right\}, \right.$$

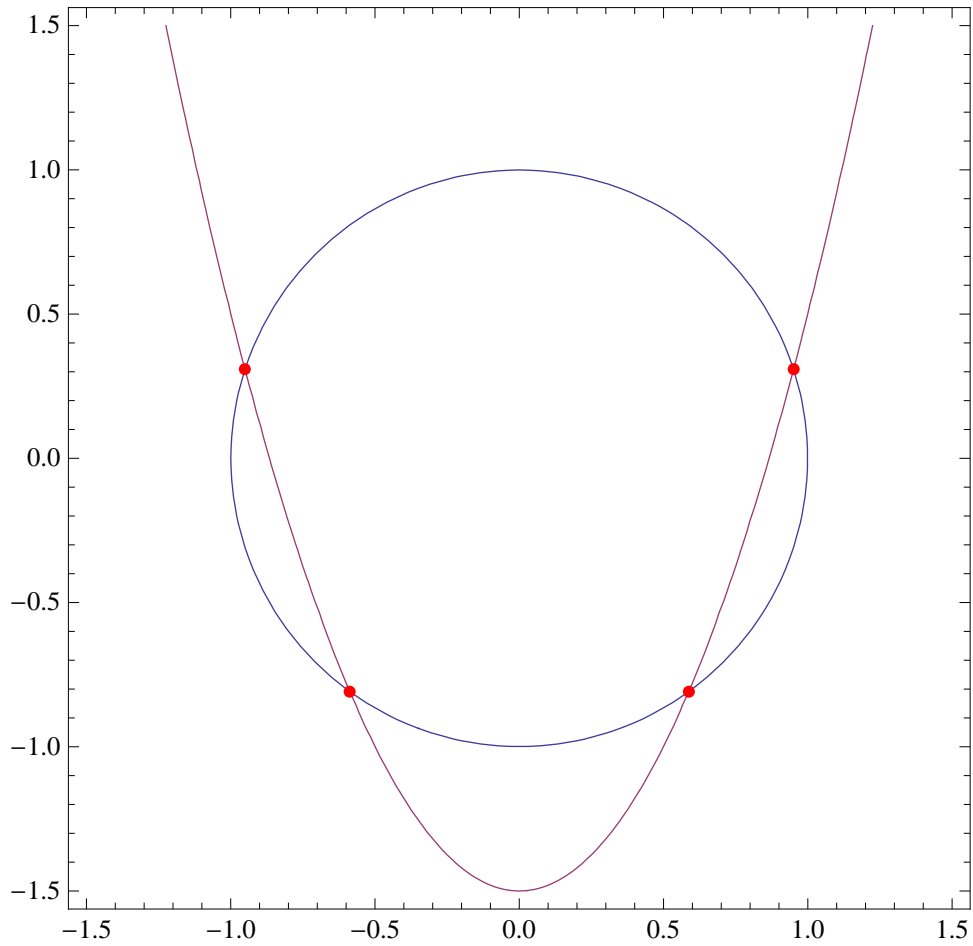
$$\left. \left\{ y \rightarrow \frac{1}{4} (-1 - \sqrt{5}), x \rightarrow \frac{1}{2} \sqrt{\frac{1}{2} (5 - \sqrt{5})} \right\}, \right.$$

$$\left. \left\{ y \rightarrow \frac{1}{4} (-1 + \sqrt{5}), x \rightarrow -\sqrt{\frac{5}{8} + \frac{\sqrt{5}}{8}} \right\}, \right.$$

$$\left. \left\{ y \rightarrow \frac{1}{4} (-1 + \sqrt{5}), x \rightarrow \sqrt{\frac{5}{8} + \frac{\sqrt{5}}{8}} \right\} \right\}$$



```
Show[ContourPlot[{x^2 + y^2 == 1, y - 2 x^2 + 3/2 == 0},  
  {x, -1.5, 1.5}, {y, -1.5, 1.5}],  
Graphics[{Red, PointSize[Medium],  
  Point[{x, y} /. pts]}]]
```



## Rekursiva funktioner

- **Exempel: Skapa en funktion som beräknar faktulteten (!) av ett tal**

$$\text{fac}(n) = \begin{cases} 1, & n = 1 \\ n * \text{fac}(n - 1), & n > 1 \end{cases}$$

```
fac[1] = 1;  
fac[n_] := n fac[n - 1];  
fac /@ {1, 2, 3, 10}
```

```
{1, 2, 6, 3 628 800}
```

- Detta är inte ett speciellt effektivt sätt att beräkna faktulteten på. Använd hellre det inbyggda kommandot `!` eller `Factorial`

```
10!  
Factorial[20]
```

```
3 628 800
```

```
2 432 902 008 176 640 000
```

## Villkor (If-satser)

- Åstadkomms med kommandot `If[villkor, värde om sant, värde om falskt]`

### ■ Exempel: maximum av två tal

```
max[x_, y_] := If[x > y, x, y];  
MapThread[max, {{1, 2, 3}, {-1, 0, 4}}]
```

```
{1, 2, 4}
```

- Kommandot finns inbyggt i *Mathematica* som `Max`

### ■ Exempel: absolutvärdet

```
abs[x_] := If[x < 0, -x, x];  
abs /@ {-1, 0, 1}
```

```
{1, 0, 1}
```

- Kommandot finns inbyggt i *Mathematica* som `Abs`

## Slingor (upprepningar)

- Slingor kan åstadkommas med kommandon som t.ex. Table, For, Do, While.

### ■ For- satser

- For-funktionen tar fyra "parametrar" For[initialisering, test, inkrementering, händelse]

```
For[i = 0, i < 4, i++, Print[i]]
```

0  
1  
2  
3

```
For[i = 1; t = x, i^2 < 10, i++, t = t^2 + i; Print[t]]
```

$1 + x^2$   
 $2 + (1 + x^2)^2$   
 $3 + (2 + (1 + x^2)^2)^2$

### ■ Do- satser

- Do-funktionen kan ta olika många parametrar. Några exempel

```
Do[Print["Test"], {4}]
```

Test  
Test  
Test  
Test

```
Do[Print[n^2], {n, 4}]
```

```
1
4
9
16
```

```
Do[Print[n], {n, -3, 5, 2}]
```

```
-3
-1
1
3
5
```

- Kan även användas med flera olika index

```
Do[Print[{i, j}], {i, 4}, {j, i - 1}]
```

```
{2, 1}
{3, 1}
{3, 2}
{4, 1}
{4, 2}
{4, 3}
```

## ■ Table

- Table-funktionen fungerar på samma sätt som Do, men ger resultatet direkt som en lista.

```
Table[n^2, {n, 10}]
```

```
{1, 4, 9, 16, 25, 36, 49, 64, 81, 100}
```

```
Table[x, {10}]
```

```
{x, x, x, x, x, x, x, x, x, x}
```

- `Table` - kan även användas för att skapa matriser

```
Table[10 i + j, {i, 4}, {j, 3}] // MatrixForm
```

$$\begin{pmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 31 & 32 & 33 \\ 41 & 42 & 43 \end{pmatrix}$$

```
Table[Prime[i], {i, 5}]
```

```
{2, 3, 5, 7, 11}
```

## ■ While- satser

- `While`-funktionen tar två parametrar: `While[test, händelse]`

```
n = 1; While[n < 4, Print[n]; n++]
```

```
1  
2  
3
```

## Använda lokala variabler i funktioner

- Problem kan uppstå om man använder samma variabelnamn till flera olika saker.
- Alla variabler är som standard globala i *Mathematica*.

### ■ Exempel: Fibonaccis talföljd, ej lokala variabler

```
Clear[f]
f[x_] = x^2;
fib[n_] :=
  Module[{},
    f[1] = f[2] = 1;
    f[i_] := f[i] = f[i - 1] + f[i - 2];
    f[n]
  ];
fib[1];
```

```
? f
```

```
Global`f
```

```
f[1] = 1
```

```
f[2] = 1
```

```
f[i_] := f[i] = f[i - 1] + f[i - 2]
```

- Funktionen `f` definieras alltså om... Inte bra!

### ■ Module

- Genom att definiera de variabler som ska vara lokala ändras deras värde inte på annat håll i programmet.
- Module-funktionen tar två "parametrar" `Module[{lokala variabler}, programkod]`

```
Clear[f]
f[x_] = x^2;
fib[n_] :=
  Module[{f, i},
    f[1] = f[2] = 1;
    f[i_] := f[i] = f[i - 1] + f[i - 2];
    f[n]
  ];
```

```
?f
```

Global`f

f[x\_] = x<sup>2</sup>

- I detta exempel är alltså f och i lokala variabler (egentligen är f en lokal funktion).

