

[Používání Maple

[> **1+1;**

2

[> **2*3;**

6

[násobení je hvězdička (*)

konec příkazu je středník (;)

konec příkazu dvojtečka (:) potlačí výstup (výpočet se provede)

[chybí-li konec příkazu :- (... ozve se chyba :

[>

[symboly:

+ (sčítání),

- (odčítání),

* (násobení),

/ (dělení),

^ nebo ** (mocniny).

help získáte

?+ pro sčítání ;-)

[>

[POZOR :

Závorky v Maple záludné

Normální jsou okrouhlé (...)

Seznam je hranatý [...]

Množina je složená { ... }

... a Maple to neplete ... :- (

[>

[Naše π je nyní pro Maple **Pi**.

[> **evalf(Pi, 3);**

3.14

[> **evalf(Pi, 10);**

3.141592654

[S e je to jinak , píšeme

exp(1)

[Při konverzi exp(1) (pravá myš, Convert to, Standard Math) to vypadá jako e

[> **exp(1);**

e

[> **evalf(exp(1), 10);**

2.718281828

[>
[>
[>

[Definujeme-li něco použijeme `:=`.

[> `x := 1;`

`x := 1`

[> `x;`

`1`

[Pak kdykoliv použijeme proměnnou, bude mít svoji hodnotu ...

[> `x+1;`

`2`

[A takhle vrátíme vše zpět,
t.j. `x` je opět "prázdné" :

[>
[>

[`Warning, premature end of input`

[> `x := 'x';`

`x := x`

[Funkci $f(x) = x + 1$ zpravidla píšeme

[> `f := x+1;`

`f := x + 1`

[> `f;`

`x + 1`

[... a dosazujeme do ni hodnoty :

[> `subs(x=1, f);`

`2`

[POZOR : $f(1)$ nic neznamená.

[>

[Klasický zápis pro definování funkce je zápis `->`

[Například :

[> `f := x -> x+1;`

`f := x → x + 1`

[> `f(x);`

`x + 1`

[> `f(1);`

`2`

[>

Je možná i tato konstrukce, z "výrazu" udělat "funkci" pomocí příkazu `unapply` :

```

[ > g := x+1;
                                      $g := x + 1$ 
[ >

[ > h := unapply(g, x);
                                      $h := x \rightarrow x + 1$ 
[ > h(1);
                                     2
[ S algebraickými výrazy jdou dělat tato kouzla :
[ simplify - zjednodušuje algebraický výraz
[ > simplify((x^2-1)/(x+1));
                                      $x - 1$ 
[ expand - roznásobí ...
[ > expand((x+1)*(x-1));
                                      $x^2 - 1$ 
[ factor - napíše jako součin ...
[ > factor(x^2-1);
                                      $(x + 1)(x - 1)$ 
[ solve - řeší systém rovnic s neznámými
[ > rovnice:={x+y=3, x-y=1};
                                      $rovnice := \{x - y = 1, x + y = 3\}$ 
[ Použili jsme "{" a "}" k definování množiny (zde rovnic)
[ > reseni:=solve(rovnice, {x,y});
                                      $reseni := \{x = 2, y = 1\}$ 
[ Zkouška
[ > subs(reseni, rovnice);
                                      $\{1 = 1, 3 = 3\}$ 
[ >
[ První řešení získáme pomocí reseni[1].
[ > reseni[1];
                                      $x = 2$ 
[ Příkazy
[ lhs-left hand side (levá strana)
[ rhs-right hand side (pravá strana)
[ dovedou rozdělit rovnost :
[ > rhs(reseni[1]);
                                     2

```

```
[ >
[ Koreny rovnice  $x^2 - 1 = 0$  jdou hledat takto
[ >
[ > koreny := RootOf(x^2-1);
[ koreny := RootOf(_Z^2 - 1)
[ > allvalues(koreny);
[ 1, -1
```

```
[ >
[ >
[ Aby bylo možno používat plot , musíme načíst
[ z knihovny procedury příkazem with(plots)
[ (PŘEDEM)
```

Pomocí příkazu **plot** získáme graf funkce f definované na intervalu $[a, b]$

plot($f, a..b, \text{volby}$)

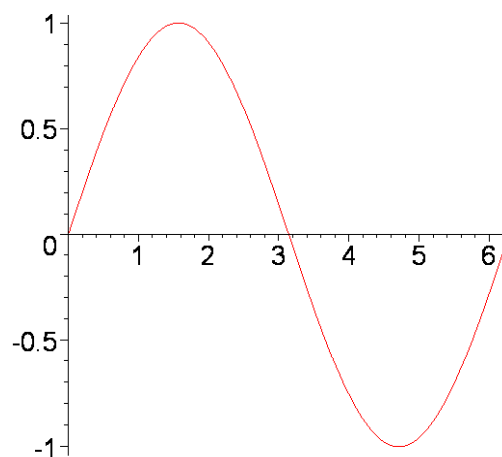
nebo

plot($f(x), x = a..b, \text{volby}$).

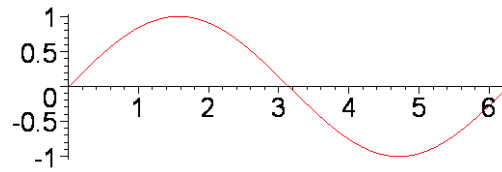
Volba **scaling=constrained** je dobrá pro správný poměr jednotek na obou osách, pokud volbu vynecháme, dělá si Maple to, co uzná za vhodné

```
>
>
Warning, premature end of input
```

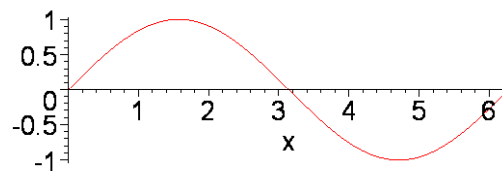
```
> f := x -> sin(x);
[ f := x → sin(x)
[ > plot(f, 0..2*Pi);
```



```
> plot(f, 0..2*Pi, scaling=constrained);
```



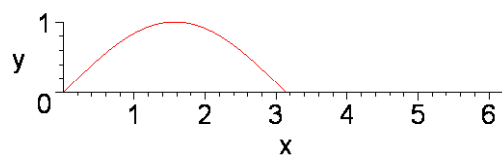
```
> plot(f(x), x=0..2*Pi, scaling=constrained);
```



T.j. : buď $f(x)$ a "x" nebo f a "nic".

Lze omezit obor hodnot:

```
> plot(f(x), x=0..2*Pi, y=0..1, scaling=constrained);
```



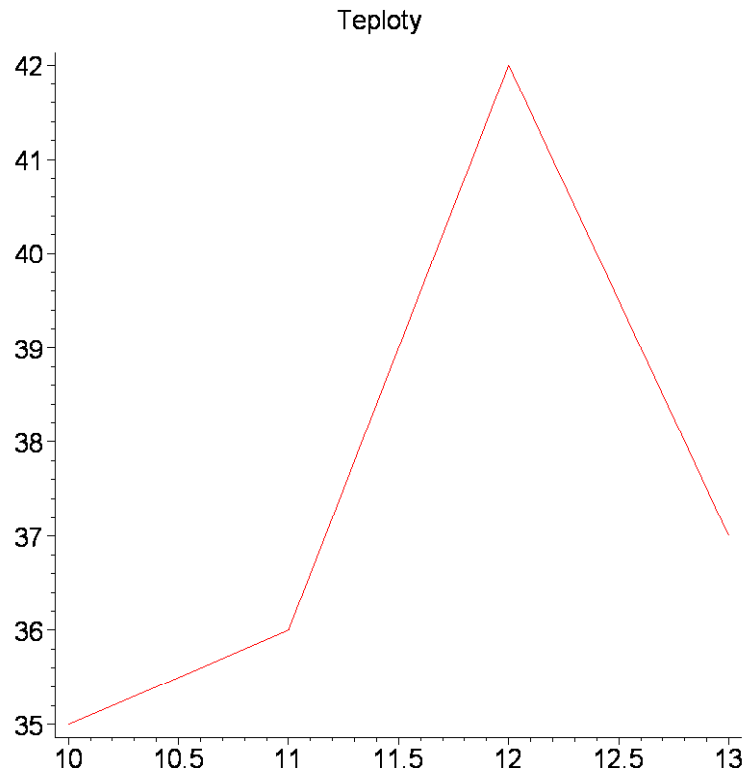
```
[ >
```

```
Seznam bodů se vytvoří :
```

```
> Puntiky := [[10,35], [11,36], [12,42], [13,37]];  
Puntiky := [[10, 35], [11, 36], [12, 42], [13, 37]]
```

```
A graf se vytvoří zase pomocí plot
```

```
> plot(Puntiky, title=`Teploty`);
```



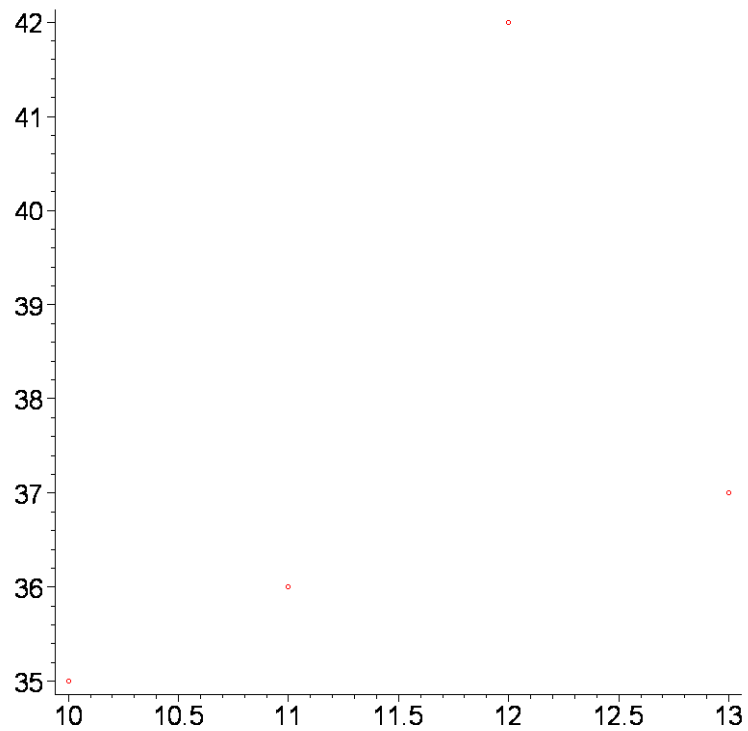
```
[ >
```

```
a volby
```

```
style=POINT  
symbol=CIRCLE  
nám udělají ...
```

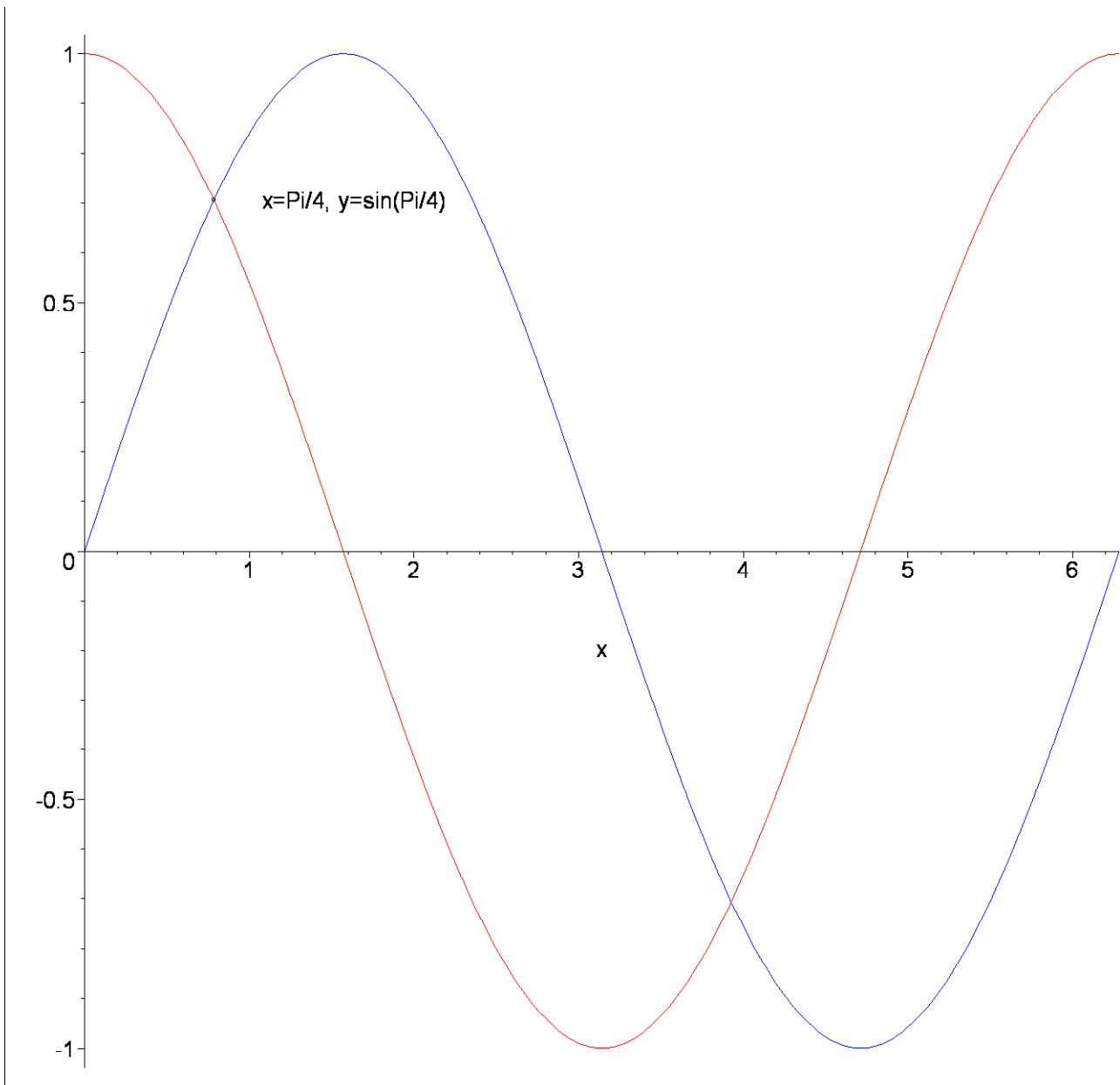
```
> plot(Puntiky, style=POINT, symbol=CIRCLE, title=`Teploty`);
```

Teploty



Grafy se mohou kombinovat do jednoho:

```
[ > plot1 := plot(f(x), x=0..2*Pi,color=blue):  
[ > plot2 := plot(cos(x), x=0..2*Pi,color=red):  
[ > plot3:=plot([[Pi/4, sin(Pi/4)]], style=POINT,  
[   symbol=CIRCLE,color=black):  
[ > plot4 := plots[textplot]([Pi/4+0.3, sin(Pi/4), `x=Pi/4,  
[   y=sin(Pi/4)`], align=RIGHT):  
[ > plots[display]({plot1, plot2, plot3,plot4});
```



Dvojtečka potlačila výstup prvních příkazů.

Příkaz `textplot` umístil písmenka na určené souřadnice, `align=RIGHT` umístilo text vpravo od souřadnice
 použily se zpětné apostrofy (```) pro ohrazení textu

`Error, missing operator or `;``

- [>
- [>
- [>
- [>