

Anleitung für Mupad von Mia Bramato

Allgemeine Befehle:

- Kommazahlen mit Punkt eingeben
- alle Rechenzeichen eingeben, also 5x eingeben als 5*x
- Wurzelzeichen gibt es nicht, statt dessen Anweisung `sqrt ()` nutzen
- Division zeichnen: /
- Um einen Befehl einzugeben eckige Klammer[
- um eine Eingabezeile zu beenden: Semikolon: ;
- zum zurücksetzen: **reset()**

Legende:

Blau= Abhängig von der jeweiligen Funktion/ Aufgabe

Rot= Bei jeder Funktion/ Aufgabe gleich

Verschiedene Mupad Befehle:

Wurzelziehen

```
[ sqrt (9)  
  3
```

Wurzel in Dezimalzahlen:

```
[ float(sqrt(9))  
  3.0
```

Rechenoperation: Division

```
[ 2/9  
   $\frac{2}{9}$ 
```

Division und Ausgabe als Kommazahl angeben (float)

```
[ float (2/9)  
  0.2222222222
```

0.2222222222

Festlegung der Anzahl der Nachkommastellen (DIGITS:=)

`DIGITS:=2`

2

`float (2/9)`

0.22

Löschen der Einstellung für Nachkommastellen

`delete DIGITS`

Funktionsgleichungen eingeben bzw. definieren:

`f1 := x-> sqrt (x)`

$x \rightarrow \sqrt{x}$

`f2 := x-> 5/x^6`

$x \rightarrow \frac{5}{x^6}$

`f3 := x-> x^6 + 4*x^4 -20*x^2-5`

$x \rightarrow x^6 + 4 \cdot x^4 - 20 \cdot x^2 - 5$

`f4 :=x->2+3*(x-3)^2`

$x \rightarrow 2 + 3 \cdot (x - 3)^2$

2

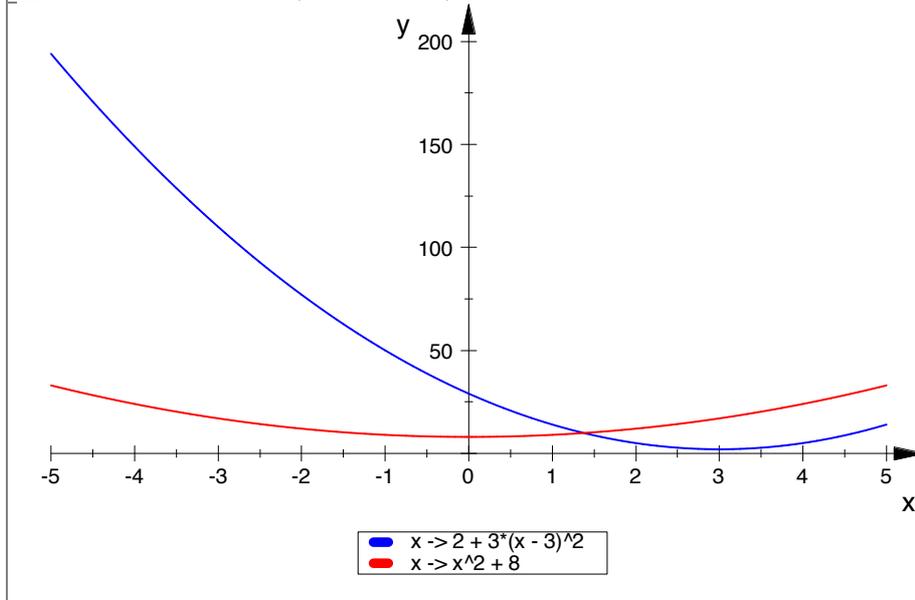
Darstellung von Funktionen

Darstellung von Funktionen einfache Darstellung:

```
f5 := x -> x^2 + 8
```

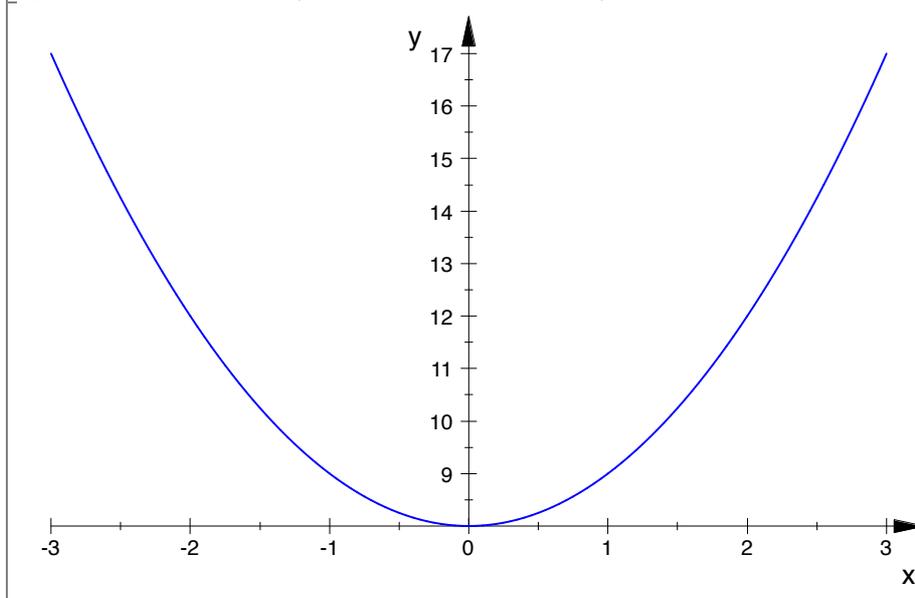
$$x \rightarrow x^2 + 8$$

```
plotfunc2d(f4, f5)
```



Darstellung mit der Vorgabe der Werte der x-Achse:

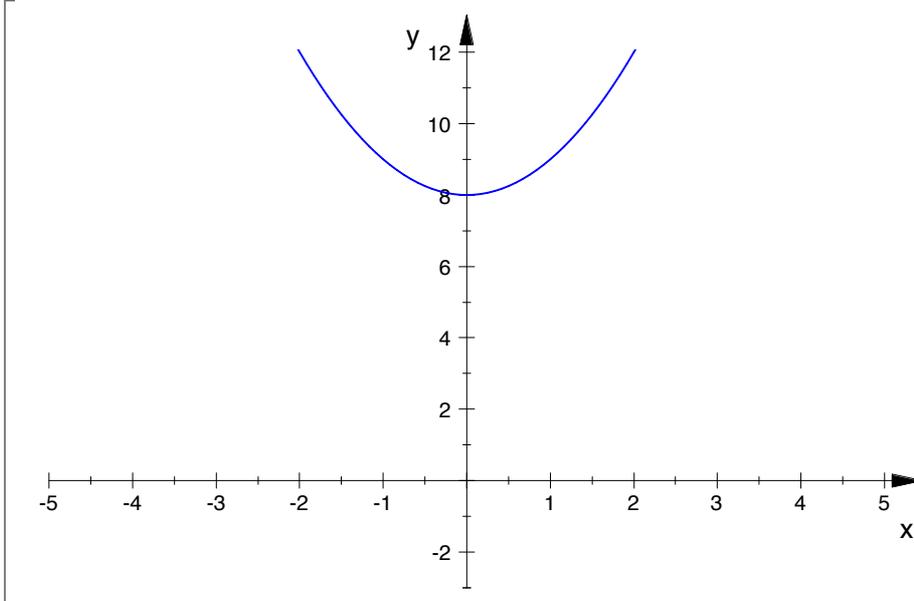
```
plotfunc2d(f5, x=-3..3)
```



Darstellung mit Vorgabe der Werte der y-Achse:

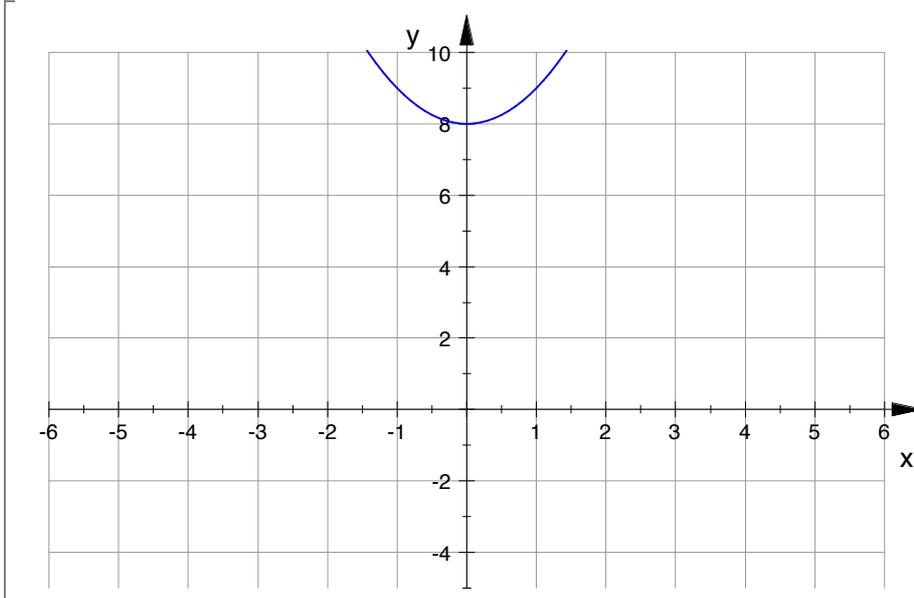
```
plotfunc2d(f5, YRange=-3..12)
```

```
plotfunc2d(f5, YRange=-3..12)
```



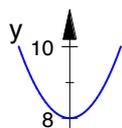
Darstellung mit Vorgabe der Werte der x-Achse und der y-Achse,
Gittereinfügen (mit dem Befehl GridVisible):

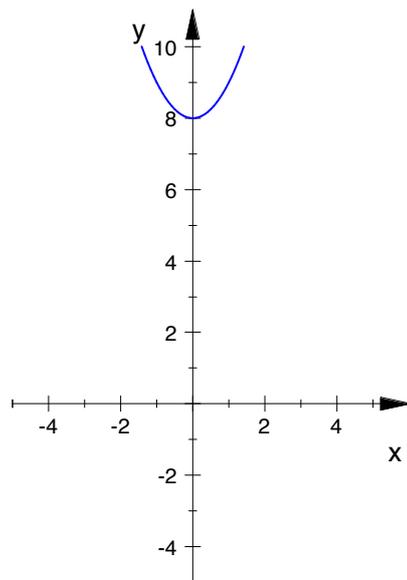
```
plotfunc2d(f5, x=-6..6, YRange=-5..10, GridVisible)
```



Darstellung mit gleichartig eingeteilter x- und y-Achse:

```
plotfunc2d(f5, x=-5..5, YRange=-5..10, Scaling=Const
```





Löschen aller vorherigen Eingaben und Speicherwerte mit reset()

```
[ reset() ]
```

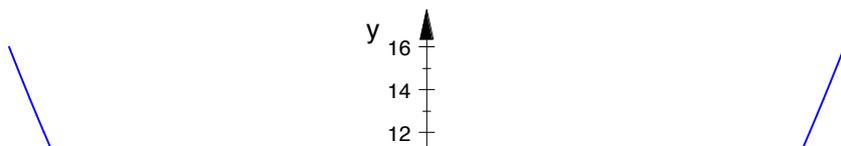
Nullstellen berechnen

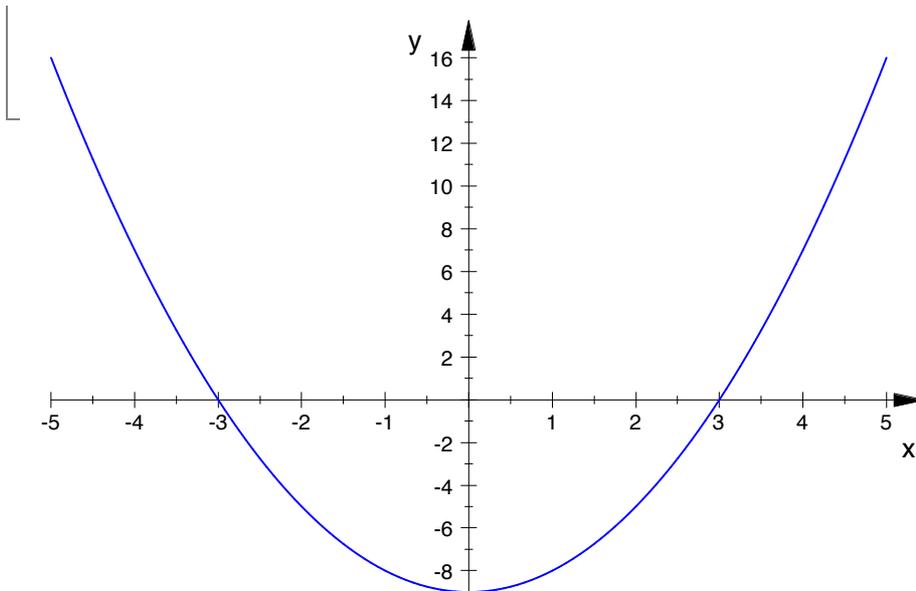
```
[ f := x -> x^2 - 9 ]
[ x -> x^2 - 9 ]
```

```
[ solve(f(x)=0) ]
[ {[x = -3], [x = 3]} ]
```

Kontrolle (muss nicht gemacht werden):

```
[ plotfunc2d(f) ]
```





```
[ f(3)
  0
```

```
[ reset()
  Beispiel
```

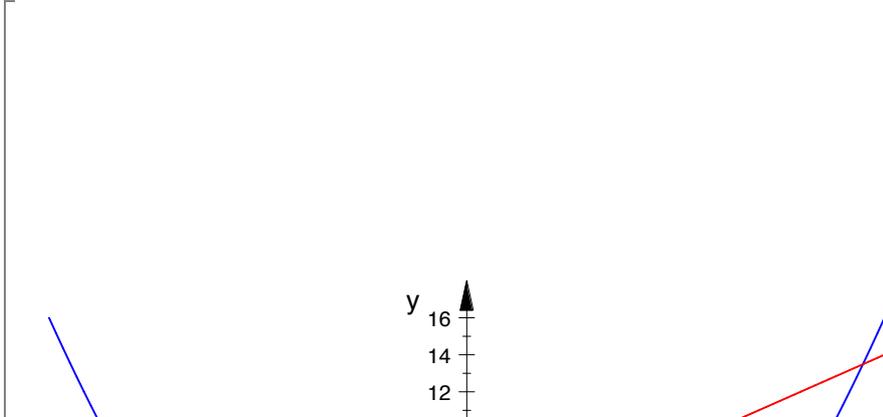
```
[ g:=x->2*x+4
  x → 2 · x + 4
```

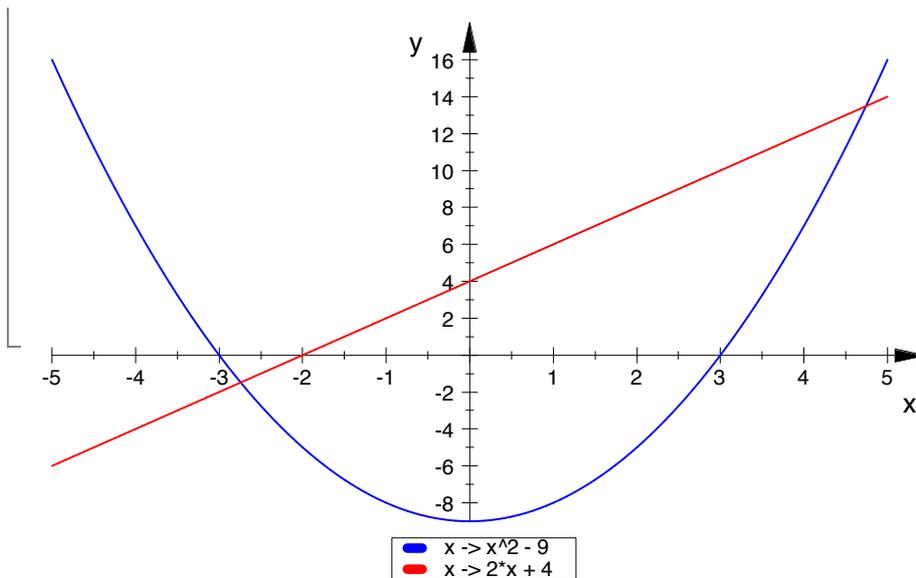
```
[ solve(g(x)=0)
  {[x = -2]}
```

Schnittpunkt berechnen

```
[ float(solve(f(x)=g(x)))
  {[x = -2.741657387], [x = 4.741657387]}
```

```
[ plotfunc2d(f,g)
```





```
float(sqrt(14)+1)
4.741657387
```

Y-Wert bestimmen vom ersten Schnittpunkt:

```
f(4.74)
13.4676
```

S(4.74/13.47)

Y-Wert bestimmen für den zweiten Schnittpunkt:

```
float(-sqrt(14)+1)
-2.741657387
```

```
f(-2.74)
-1.4924
```

S(-2.74/-13.49)

Ergebnis in Dezimalzahlen:

```
float(solve(f(x)=g(x)))
```

```
[ float(solve(f(x)=g(x)))
  {[x = -2.741657387], [x = 4.741657387]}
]
```

Ausmultiplizieren:

```
[ k:=x-> 3*(x+4)^2-13
  x -> 3*(x+4)^2 - 13
]
```

```
[ expand(k)
  x -> 3*(x+4)^2 - 13
]
```

Funktionswert an einer bestimmten Stelle: (im Beispiel Stelle: 17)

```
[ k(17)
  1310
]
```

Von Funktionswert auf Stelle schließen: (im Beispiel 1310)

```
[ solve(k(x)=1310, X)
  {
    Ø if (x+4)^2 ≠ 441
    C if (x+4)^2 = 441
  }
]
```

```
[
```