Muster in Mma , WQ, FS 2013

```
m1[f[x_]] := f[x + 1]

list1 = {f[x], f[2, 3], g[y], f[1 / x]}
```

$\{f[x], f[2, 3], g[y], f[\frac{1}{x}]\}$

```
Map[m1, list1]
```

$\{f[1 + x], m1[f[2, 3]], m1[g[y]], f[1 + \frac{1}{x}]\}$

```
(* next *)

m2[_[y_]] := y / x

Map[m2, list1]
```

$\{1, m2[f[2, 3]], \frac{y}{x}, \frac{1}{x^2}\}$

```
(*  ist alles auch mehrdimensional moeglich *)

Clear[f]
f[x_, y_] := Sqrt[x^2 + y^2]

{f[a, b], f[3, 4]}
```

$\{\sqrt{a^2 + b^2}, 5\}$

```
Clear[f]
f[{x_, y_}] := {y, x}

Map[f, { {1}, {2, 3}, {4, 5, 6}}]

{f[{1}], {3, 2}, f[{4, 5, 6}]}

(* Vorangehende Definition der Fibonacci-
 Zahlen war nicht gegen 'falsches' Einsetzen gefeit *)

fib[0] = fib[1] = 1;
fib[n_] := fib[n - 1] + fib[n - 2]

fib[-3]
```

$RecursionLimit::reclim : Recursion depth of 1024 exceeded. ≫

```
fib[5.5]
```

$RecursionLimit::reclim : Recursion depth of 1024 exceeded. ≫

```
(* Ausweg: Alles 'Falsche' ausschliessen *)
Clear[fib]

fib[0] = fib[1] = 1;
fib[n_? (IntegerQ[#] ⋀ NonNegative[#] &)] := fib[n - 1] + fib[n - 2]
```

```
fib[-3]
```
fib[-3]

```
fib[5.5]
```
fib[5.5]

```
(* Muster bedingter Vorschriften, Condition  /;  *)

ourBinom[n_, k_] := n! / k! / (n - k)!  /;
   IntegerQ[n] ⋀ IntegerQ[k] ⋀ n ≥ 0 ⋀ k ≥ 0

list2 = {7, -5, Sqrt[12], 5.5, z}
```
$\{7, -5, 2\sqrt{3}, 5.5, z\}$

```
Map[ourBinom[#, 2] &, list2]
```
$\{21, \text{ourBinom}[-5, 2], \text{ourBinom}[2\sqrt{3}, 2], \text{ourBinom}[5.5, 2], \text{ourBinom}[z, 2]\}$

```
(* Wichtige XxxQ-Funktionen:
  EvenQ     OddQ    ListQ    IntegerQ    MatrixQ
   allgemeine Abfrage mit: *)

? *Q
```

Typmuster x_Typ
Diese koennen mit konditionalen Mustern kombiniert werden

```
Clear[ourBinom]

ourBinom[n_, k_Integer?NonNegative] := Product[(n - i - 1) / i, {i, 1, k}]

Map[ourBinom[#, 2] &, list2]
```
$\{21, 28, \frac{1}{2}(-3+2\sqrt{3})(-2+2\sqrt{3}), 4.375, \frac{1}{2}(-3+z)(-2+z)\}$

```
(*  Nun gut, wie mathematisch vernuenftig das fuer n ist,
ist eine andere Frage. Fuer k kontrolliert es gut: *)

Map[ourBinom[#, -5] &, list2]
```
$\{\text{ourBinom}[7, -5], \text{ourBinom}[-5, -5],$
$\text{ourBinom}[2\sqrt{3}, -5], \text{ourBinom}[5.5, -5], \text{ourBinom}[z, -5]\}$

```
(* next *)

Clear[f]

f[x_Integer] := x

f /@ list2
```
$\{7, -5, f[2\sqrt{3}], f[5.5], f[z]\}$

```
(* Aber *)
```

```
Clear[f]

f[x_Real] := x

f /@ list2
```

$\{f[7], f[-5], f[2\sqrt{3}], 5.5, f[z]\}$

```
(* Die Integer sind keine 'Reals'  und
 die Wurzel auch nicht !! 1.Ausweg: *)

f[x_Integer] := x

f /@ list2
```

$\{7, -5, f[2\sqrt{3}], 5.5, f[z]\}$

```
? f
```

Global`f

```
f[x_Real] := x

f[x_Integer] := x

(* nun fuer alle rellen Zahlen :  *)

f[x_?NumericQ] := f[N[x]]

f /@ list2
{7, -5, 3.4641, 5.5, f[z]}

(* next

 Variable Parameteranzahl :  in Mma
                             Plus   Times    Min   Max

   mit Platzhaltern  __  fuer ein oder mehrere Argumente
   mit              ___  fuer null  -- " --             *)

relSum[a_, b_] := (a + b) / (1 + a * b)

relSum[a_, b_, c__] := relSum[a, relSum[b, c]] // Together

relSum[x, y, z]
```

$$\frac{x + y + z + x\,y\,z}{1 + x\,y + x\,z + y\,z}$$

```
relSum[0.3, 0.4, 0.5, 0.6]

0.962264

relSum[a, 1, b]

1

(*  next :   Wiederholte Muster   /.  und   //.    *)
```

```
a^2 + 2 a b^2 /. {a → b^2 + 1, b → 2}
```

$8\left(1+b^2\right)+\left(1+b^2\right)^2$

```
a^2 + 2 a b^2 //. {a → b^2 + 1, b → 2}
```

65

```
(*  next :  Teillisten  und  Muster  *)
```

```
list3 = {1, a, b^c, d^3, e + f, Sqrt[3]}
```

$\left\{1, a, b^c, d^3, e+f, \sqrt{3}\right\}$

```
Select[list3, MatchQ[#, _^_] & ]
```

$\left\{b^c, d^3, \sqrt{3}\right\}$

```
u = 12 (1 + x) ^5 // Expand
```

$12+60\,x+120\,x^2+120\,x^3+60\,x^4+12\,x^5$

```
Cases[u, 12 _^_]
```

$\left\{12\,x^5\right\}$

```
Cases[u, _ x^4]
```

$\left\{60\,x^4\right\}$

```
Cases[u, _ __]
```

$\left\{60\,x, 120\,x^2, 120\,x^3, 60\,x^4, 12\,x^5\right\}$

```
Select[u, MatchQ[#, _ __] &]
```

$60\,x+120\,x^2+120\,x^3+60\,x^4+12\,x^5$

```
List @@ %
```

$\left\{60\,x, 120\,x^2, 120\,x^3, 60\,x^4, 12\,x^5\right\}$

```
Cases[list3, _Symbol]
```

{a}

```
Count[list3, _Symbol]
```

1

```
MemberQ[list3, _Times]
```

False

```
(*  Eigene Operatoren definieren,
Bsp. relativistische Summe von Geschwindigkeiten *)
```

```
Clear[CirclePlus]
CirclePlus[a_, b_] := (a + b) / (1 + a * b)

CirclePlus[a_, b_, c__] := CirclePlus[a, CirclePlus[b, c]] // Together

x ⊕ y ⊕ z        (*  Esc C+ Esc  *)
```

$$\frac{x + y + z + x\,y\,z}{1 + x\,y + x\,z + y\,z}$$

```
0.3 ⊕ 0.4 ⊕ 0.5 ⊕ 0.6
```

0.962264

```
a ⊕ 1 ⊕ b
```

1.

```
2 ⊕ 2
```

$$\frac{4}{5}$$