

Einführung in Sage

Zusammenfassung Einheit 02

Hinweis: Textbausteine mit `<name>` weisen darauf hin, das anstatt des Ausdrucks eine passende Variable eingefügt werden muss.

Grundlagen

- Bezeichner und Wert: `<bezeichner> = <wert>`

- Typ eines Objekts- `type()`

```
type(<var>)
```

- Ausgabe- `print`

```
print ('Text %<format> und %<format> .. ' %  
      (x,y,...))
```

Dictionaries- `dictionaries`

- Deklarieren eines Dictionaries:

```
d = {<Index1>:<Wert1>,<Index2>:<Wert2>,...}
```

- Zugriff auf Index:

```
d[<Index>]
```

Funktionen und Abfragen

- anonyme Funktion- `lambda`

```
lambda <parameter_list>: <expression>
```

- Schleifen- `for`

```
for <indexvar> in <expr>/<list>:  
    <Code-Block>  
else:  
    <Code-Block>
```

- Abfragen- `if`

```
if <boolean expr>:  
    <Code-Block>
```

Ausdrücke

- Sortieren nach `<x>`- `collect()`

```
<expr>.collect(<x>)
```

- Potenzgesetze- `combine()`

```
<expr>.combine()
```

- Ausmultiplizieren- `expand()`

```
<expr>.expand()
```

- Faktorisieren- `factor()`

```
<expr>.factor()
```

- Partialbruch-Zerlegung- `partial_fraction()`

```
<expr>.partial_fraction()
```

- Vereinfachen

- trigonometrisch- `simplify_trig()`

```
<expr>.simplify_trig()
```

- rational- `simplify_rational()`

```
<expr>.simplify_rational()
```

- log/ln/exp- `simplify_radical()`

```
<expr>.simplify_radical()
```

- Nutzung der Fakultät- `simplify_factorial()`

```
<expr>.simplify_factorial()
```

- alle Vereinfachungen- `simplify_full()`

```
<expr>.simplify_full()
```

Gleichungen, Vergleiche, Logik

- Gleichungssysteme analytisch lösen mit `solve()`:

```
solve(<gleichungen>,<variablen>,<optionen>)
```

- Gleichungssysteme numerisch lösen mit `find_root()`:

```
<expr>.find_root(<lowerbound>,<upperbound>)
```

- Wahrheitswert für die Gleichung- `boolean`

```
bool(<Ausdruck/Gleichung>)
```

- Wahrheitswerte: `True`, `False`

- Logische Verknüpfungen: `and`, `or`, `not`