

Python-Modul units

Einführung und Anleitung – eine Dokumentation mittels PythonT_EX

Günter Partosch*

Version 1.11.2, 2018-08-30

* Gunter.Partosch@hrz.uni-giessen.de

Inhaltsverzeichnis

1	Einleitung und Übersicht	1
1.1	Übersicht	1
1.2	Start	2
1.3	Einleitende Beispiele	3
1.4	Informationen zum Modul	6
2	Basisklasse L1	16
2.1	Übersicht	16
2.2	Methoden der Klasse	17
2.3	Maßeinheiten und Instanzen	18
2.4	Operatoren	22
2.5	Eigenschaften der L1-Instanzen	23
2.6	Konvertierung	24
3	Basisklasse L2	26
3.1	Übersicht	26
3.2	Methoden der Klasse	28
3.3	Maßeinheiten und Instanzen	29
3.4	Operatoren	31
3.5	Eigenschaften der L2-Instanzen	33
3.6	Konvertierung	34
4	Basisklasse L3	36
4.1	Übersicht	36
4.2	Methoden der Klasse	38
4.3	Maßeinheiten und Instanzen	39
4.4	Operatoren	41
4.5	Eigenschaften der L3-Instanzen	43
4.6	Konvertierung	44
5	Basisklasse T1	45
5.1	Übersicht	45
5.2	Methoden der Klasse	47
5.3	Maßeinheiten und Instanzen	48
5.4	Operatoren	50
5.5	Eigenschaften der T1-Instanzen	52
5.6	Konvertierung	53
6	Basisklasse T2	54
6.1	Übersicht	54
6.2	Methoden der Klasse	56

6.3	Maßeinheiten und Instanzen	57
6.4	Operatoren	59
6.5	Eigenschaften der T2-Instanzen	61
6.6	Konvertierung	62
7	Basisklasse M	63
7.1	Übersicht	63
7.2	Methoden der Klasse	65
7.3	Maßeinheiten und Instanzen	66
7.4	Operatoren	69
7.5	Eigenschaften der M-Instanzen	71
7.6	Konvertierung	71
8	Basisklasse TT	73
8.1	Übersicht	73
8.2	Methoden der Klasse	75
8.3	Maßeinheiten und Instanzen	76
8.4	Operatoren	77
8.5	Eigenschaften der TT-Instanzen	79
8.6	Konvertierung	80
9	Basisklasse G	81
9.1	Übersicht	81
9.2	Methoden der Klasse	82
9.3	Maßeinheiten und Instanzen	83
9.4	Operatoren	87
9.5	Eigenschaften der G-Instanzen	89
9.6	Konvertierung	90
10	Basisklasse N	91
10.1	Übersicht	91
10.2	Methoden der Klasse	92
10.3	Maßeinheiten und Instanzen	93
10.4	Operatoren	94
10.5	Eigenschaften der N-Instanzen	96
11	Abgeleitete Klasse U	97
11.1	Übersicht	97
11.2	Methoden der Klasse	99
11.3	Maßeinheiten und Instanzen	100
11.4	Operatoren	101
11.5	Eigenschaften der U-Instanzen	103
11.6	Konvertierung	104
12	Abgeleitete Klasse V	105
12.1	Übersicht	105
12.2	Methoden der Klasse	107
12.3	Maßeinheiten und Instanzen	108

12.4 Operatoren	110
12.5 Eigenschaften der V-Instanzen	112
12.6 Konvertierung	113
13 Abgeleitete Klasse B	115
13.1 Übersicht	115
13.2 Methoden der Klasse	116
13.3 Maßeinheiten und Instanzen	117
13.4 Operatoren	120
13.5 Eigenschaften der B-Instanzen	122
13.6 Konvertierung	123
14 Abgeleitete Klasse F1	125
14.1 Übersicht	125
14.2 Methoden der Klasse	126
14.3 Maßeinheiten und Instanzen	127
14.4 Operatoren	130
14.5 Eigenschaften der F1-Instanzen	132
14.6 Konvertierung	133
15 Abgeleitete Klasse W	135
15.1 Übersicht	135
15.2 Methoden der Klasse	136
15.3 Maßeinheiten und Instanzen	137
15.4 Operatoren	140
15.5 Eigenschaften der W-Instanzen	142
15.6 Konvertierung	143
16 Abgeleitete Klasse P	145
16.1 Übersicht	145
16.2 Methoden der Klasse	147
16.3 Maßeinheiten und Instanzen	148
16.4 Operatoren	150
16.5 Eigenschaften der P-Instanzen	152
16.6 Konvertierungen	153
17 Kombinierte Anwendungen	155
17.1 Kombinierte Operationen	155
17.2 Globale Methoden	157
Index	158

Tabellenverzeichnis

1.1	Globale Methoden zum Generieren von Instanzen	10
1.2	Globale Hilfsmethoden	11
1.3	Globale Methoden, in denen Klassen kombiniert werden	13
1.4	Kurzbezeichner (Aliase) für einige Methoden und Klassen	14
2.1	Methoden der Klasse L1	18
2.2	Klasse L1, arithmetische Operatoren und rechtsseitige Operanden	23
3.1	Methoden der Klasse L2	28
3.2	Klasse L2, arithmetische Operatoren und rechtsseitige Operanden	32
4.1	Methoden der Klasse L3	38
4.2	Klasse L3, arithmetische Operatoren und rechtsseitige Operanden	42
5.1	Methoden der Klasse T1	47
5.2	Klasse T1, arithmetische Operatoren und rechtsseitige Operanden	51
6.1	Methoden der Klasse T2	56
6.2	Klasse T2, arithmetische Operatoren und rechtsseitige Operanden	60
7.1	Methoden der Klasse M	65
7.2	Klasse M, arithmetische Operatoren und rechtsseitige Operanden	70
8.1	Methoden der Klasse TT	75
8.2	Klasse TT, arithmetische Operatoren und rechtsseitige Operanden	78
9.1	Methoden der Klasse G	82
9.2	Klasse G, arithmetische Operatoren und rechtsseitige Operanden	88
10.1	Methoden der Klasse N	92
10.2	Klasse N, arithmetische Operatoren und rechtsseitige Operanden	95
11.1	Methoden der Klasse U	99
11.2	Klasse U, arithmetische Operatoren und rechtsseitige Operanden	102
12.1	Methoden der Klasse V	107
12.2	Klasse V, arithmetische Operatoren und rechtsseitige Operanden	112
13.1	Methoden der Klasse B	117
13.2	Klasse B, arithmetische Operatoren und rechtsseitige Operanden	121
14.1	Methoden der Klasse F1	127
14.2	Klasse F1, arithmetischen Operatoren und rechtsseitige Operanden	131

Tabellenverzeichnis

15.1 Methoden der Klasse \mathbb{W}	137
15.2 Klasse \mathbb{W} , arithmetische Operatoren und rechtsseitige Operanden	141
16.1 Methoden der Klasse \mathbb{P}	147
16.2 Klasse \mathbb{P} , arithmetische Operatoren und rechtsseitige Operanden	152
17.1 Zulässige Operationen mit dem Operator $*$	155
17.2 Zulässige Operationen mit dem Operator $/$	155

1 Einleitung und Übersicht

Die vorliegende Anleitung beschreibt den Umgang mit dem Python-Modul `units.py`.¹ Die aktuelle Beschreibung gibt es bei <http://www.staff.uni-giessen.de/partosch/unterlagen/units.pdf>.

1.1 Übersicht

Das Python-Modul `units.py` ermöglicht das Rechnen mit klassischen physikalischen Einheiten (wie `m`, `kg`, `s`, `K`, ...).

- Dazu stellt das Modul als Basis mehrere Python-Klassen mit spezifischen Methoden und Eigenschaften bereit:
 - ◇ L1 (Länge): mit den Maßeinheiten `aa`, `cm`, `dm`, `km`, `m`, `mm`, `my`, `nm`, ..., den amerikanischen Längenmaßen `ft`, `inch`, `mi`, `sm`, `yd`, `Zoll` und den Druckermäßen `bp`, `pt`, `dd`, `cc`, `pc`, `sp`, `ze` → Abschnitt 2 auf Seite 16
 - ◇ L2 (Fläche): mit den Maßeinheiten `aa2`, `ar`, `cm2`, `dm2`, `ha`, `km2`, `m2`, `mm2`, `my2`, ... und den amerikanischen Maßen `ft2`, `inch2`, `mi2`, `sm2`, `yd2` → Abschnitt 3 auf Seite 26
 - ◇ L3 (Volumen): mit den Maßeinheiten `cm3`, `dm3`, `km3`, `m3`, `mm3`, `l`, `liter`, `cl`, `dl`, `hl`, `ml`, ... und den amerikanischen Maßen `inch3`, `ft3`, `gal`, `bb1`, `pint`, `yd3` → Abschnitt 4 auf Seite 36
 - ◇ M (Masse/Gewicht): mit den Maßeinheiten `t`, `kg`, `g`, `mg`, `myg`, ..., `pfd`, `dz`, `Ztr`, ... und amerikanischen Gewichtsmaßen → Abschnitt 7 auf Seite 63
 - ◇ T1 (Zeit): mit den Maßeinheiten `ns`, `mys`, `ms`, `s`, `h`, `minute`, `d`, `week`, `mon`, `y`, ... → Abschnitt 5 auf Seite 45
 - ◇ T2 (Zeit²): mit den Maßeinheiten `ns2`, `mys2`, `ms2`, `s2`, `h2`, `minute2`, `d2`, `week2`, `mon2`, `y2`, ... → Abschnitt 6 auf Seite 54
 - ◇ TT (Temperatur): mit den Maßeinheiten `C`, `F`, `K`, `R` → Abschnitt 8 auf Seite 73
- Ergänzt wird das System durch eine Reihe spezieller Klassen, in denen die Möglichkeiten der Basis-Klassen kombiniert werden:
 - ◇ G (allgemeine Klasse zum Kombinieren); ohne spezifische eigene Maßeinheiten → Abschnitt 9 auf Seite 81
 - ◇ V (Geschwindigkeit); ohne spezifische eigene Maßeinheiten → Abschnitt 12 auf Seite 105
 - ◇ B (Beschleunigung); ohne spezifische eigene Maßeinheiten → Abschnitt 13 auf Seite 115
 - ◇ F1 (Kraft); ohne spezifische eigene Maßeinheiten → Abschnitt 14 auf Seite 125
 - ◇ W (Arbeit/Energie); ohne spezifische eigene Maßeinheiten → Abschnitt 15 auf Seite 135
 - ◇ P (Leistung); ohne spezifische eigene Maßeinheiten → Abschnitt 16 auf Seite 145

¹ Ursprünglich [2017] sollte anhand eines ausführlicheren Beispiels lediglich gezeigt werden, wie in Python Klassen und Subklassen entwickelt und eingesetzt werden können. Zusätzlich sollte demonstriert werden, wie elegant Python-Konstrukte mit Hilfe von `PythonTeX` dargestellt werden können. Die jeweils neueste Version des Moduls finden Sie unter <http://www.staff.uni-giessen.de/partosch/unterlagen/units.zip>

- Daneben gibt es noch:
 - ◇ **N** (Hilfsklasse zum linksseitigen Multiplizieren mit einem Skalar) → Abschnitt 10 auf Seite 91
 - ◇ **U** (Uhrzeiten) → Abschnitt 11 auf Seite 97
- Mit Hilfe der obigen Klassen können Instanzen generiert werden, die physikalische Größen (jeweils bestehend aus Maßzahl und Maßeinheit) darstellen und sich ähnlich wie Variablen in einer herkömmlichen Programmiersprache verhalten:
 1. Generierung mit Maßzahl und Maßeinheit
 2. Zuweisungen
 3. Ausdrücke mit arithmetischen Operatoren (ggf. mit Klammern)
 4. Vergleiche
 5. Konvertierungen

1.2 Start

Damit Sie die Möglichkeiten des Moduls nutzen können, muss es in der von Ihnen eingesetzten Python-Benutzeroberfläche (bei mir IDLE) bekannt und verfügbar gemacht werden.

- Dazu müssen Sie das Modul zunächst per Download beschaffen:
<http://www.staff.uni-giessen.de/partosch/unterlagen/units.zip>
- Entpacken Sie diese **.zip**-Datei in ein geeignetes Verzeichnis, das von Python erreicht werden kann. Bei mir ist das **D:/Python**.
- Anschließend sollten Sie die Python-Benutzerfläche informieren, wo das Modul **units.py** zu finden ist. Das geschieht am einfachsten dadurch, dass die Umgebungsvariable **PYTHONPATH** entsprechend erweitert wird. Temporär können Sie dazu unter Windows in der Kommandozeile

```
set PYTHONPATH=D:/Python; bisheriger Python-Pfad
```

eingeben oder über den Weg *Start > Systemsteuerung > System > Erweiterte Systemeinstellungen > Umgebungsvariablen...* die Umgebungsvariable **PYTHONPATH** dauerhaft ändern.

- Falls Sie keinen direkten Zugang zur Umgebungsvariablen **PYTHONPATH** haben, geben Sie in der Python-Benutzerfläche

```
>>> import sys; sys.path.append('D:/Python'); # Python-Pfad ergänzen
```

ein.

- Danach können Sie das Modul in der Python-Benutzeroberfläche laden:

```
>>> from units import * # alle Methoden/Eigenschaften ohne Präfix importieren
```

Danach sind alle Methoden und Eigenschaften des Moduls verfügbar. Ein Test zeigt

```
>>> modul; version; date
'units.py'
'9.16.2'
'2018-07-29'
```


1.3 Einleitende Beispiele

Für die Darstellung von Instanzen gelten im Folgenden zunächst einige Voreinstellungen, die durch einen Aufruf der globalen Methode `globalInfo('V')` (oder `globalInfo('E')`) aufgelistet werden können:

```
>>> globalInfo('V')
Globale Variablen/Eigenschaften
-----
eps      1e-16      für Vergleiche auf Gleichheit: ganz klein
rndg     4          für Rundung: Nachkommastellen
science False      Darstellung von Instanzen in Science Notation
trennz   1          Trennzeichen zw. Maß und Maßeinheit in __str__/_repr__

Diese Eigenschaften können mit Hilfe der globalen Methode setVar geändert
werden.

Außerdem:
modul    units.py   Name des Moduls
version  9.16.2     aktuelle Version des Moduls
date     2018-07-29 Datum der letzten Änderung
-----
```

Generierung von Instanzen mit Maßzahl und Maßeinheit

```
>>> L1(3,m)                                     # L1-Instanz: 3 m
L1(3 m)
>>> L2(2.5,ha)                                 # L2-Instanz: 2.5 ha
L2(2.5 ha)
>>> L3(1.3,liter)                             # L3-Instanz: 1.3 l
L3(1.3 liter)
>>> M(2.5,t)                                   # M-Instanz: 2.5 t
M(2.5 t)
>>> T1(3,minute)                              # T1-Instanz: 3 min
T1(3 minute)
>>> T2(2.5,s2)                                # T2-Instanz: 2.5 s2
T2(2.5 s2)
>>> TT(300,K)                                 # TT-Instanz: 300 K
TT(300 K)
>>> newton(200)                               # F1-Instanz: 200 N
F1(M(200 kg), B(L1(1 m), T2(1 s2)))
>>> joule(100)                                # W-Instanz: 100 J
W(F1(M(100 kg), B(L1(1 m), T2(1 s2))), L1(1 m))
>>> watt(2300)                                 # P-Instanz: 2300 W
P(W(F1(M(2300 kg), B(L1(1 m), T2(1 s2))), L1(1 m)), T1(1 s))
>>> meterS(300)                               # V-Instanz: 300 m/s
V(L1(300 m), T1(1 s))
```

1 Einleitung und Übersicht

```
>>> meterS2(2.5) # B-Instanz: 2.5 m/s2
B(L1(2.5 m), T2(1 s2))
>>> G(L3(200,liter), div, L2(1,m2)) # G-Instanz: 200 l/m2
G(L3(200 liter) / (L2(1 m2)))
>>> G(meterS(300),mul,T1(30,s)) # G-Instanz: 300 m/s * 30 s
G(V(L1(300 m), T1(1 s)) * (T1(30 s)))
```

Zuweisungen

```
>>> z11 = L1(3,m); z12 = L1(4,m); z21 = L2(4,m2); z22 = L2(5,km2) # Zuweisung
>>> z11, z12, z21, z22
(L1(3 m), L1(4 m), L2(4 m2), L2(5 km2))
>>> z23 = z21; z24 = z22 # Zuweisung
>>> z13, z14 = z11, z12 # Zuweisung
>>> z13, z14, z23, z24
(L1(3 m), L1(4 m), L2(4 m2), L2(5 km2))
```

Ausdrücke mit arithmetischen Operatoren

```
>>> # Addition
>>> z11 + L1(5,km) # L1 + L1
L1(5003.0 m)
>>> L2(2,ha) + z22 # L2 + L2
L2(5020000.0 m2)
>>> T1(2,minute) + T1(3,h) # T1 + T1
T1(10920 s)
>>> M(2,g) + M(3,kg) # M + M
M(3.002 kg)
```

```
>>> # Subtraktion
>>> z11 - L1(5,km) # L1 - L1
L1(-4997.0 m)
>>> L2(2,ha) - z22 # L2 - L2
L2(-4980000.0 m2)
>>> T1(2,minute) - T1(3,h) # T1 - T1
T1(-10680 s)
>>> M(2,g) - M(3,kg) # M - M
M(-2.998 kg)
```

```
>>> # Vorzeichen
>>> -z11 # -(L1)
L1(-3 m)
>>> +z21 # +(L2)
L2(4 m2)
>>> -L3(3,liter) # -(L3)
L3(-3 liter)
>>> +T2(3,h2) # +(T2)
```

1 Einleitung und Übersicht

```
T2(3 h2)
```

```
>>> # Multiplikation
```

```
>>> L1(3,m) * L1(5,m)
```

```
# L1 * L1
```

```
L2(15 m2)
```

```
>>> T1(3,s) * T1(2, minute)
```

```
# T1 * T1
```

```
T2(360 s2)
```

```
>>> L2(2,m2) * (L1(1.5,m) + L1(2,cm))
```

```
# L2 * (L1 + L1)
```

```
L3(3.04 m3)
```

```
>>> # Division
```

```
>>> z12 / L1(2,m)
```

```
# L1 / L1
```

```
2.0
```

```
>>> L1(300,m) / T1(10,s)
```

```
# L1 / T1
```

```
V(L1(30.0 m), T1(1 s))
```

```
>>> L3(2.5,liter) / L2(200,cm2)
```

```
# L3 / L2
```

```
L1(0.125 m)
```

```
>>> # Exponentiation
```

```
>>> z11 **2
```

```
# L1 ** 2
```

```
L2(9 m2)
```

```
>>> T1(10,s) ** 2
```

```
# T1 ** 2
```

```
T2(100 s2)
```

```
>>> L1(300,m) ** 3
```

```
# L1 ** 3
```

```
L3(27000000 m3)
```

```
>>> T2(64,s2) ** 0.5
```

```
# T2 ** 0.5
```

```
T1(8.0 s)
```

Vergleiche

```
>>> z11 < z12
```

```
# L1 < L1
```

```
True
```

```
>>> z22 > z21
```

```
# L2 > L2
```

```
True
```

```
>>> z12 >= L1(0,m)
```

```
# L1 >= L1
```

```
True
```

```
>>> z22 <= L2(10,m2)
```

```
# L2 <= L2
```

```
False
```

```
>>> (L1(1, m) <= z11) and (z11 <= L1(5, m))
```

```
# z11 im Intervall [1m, 5m]
```

```
True
```

```
>>> not ((z11 == L1(1, m)) or (z11 == L1(5, m))) # z11 weder 1m noch 5m
```

```
True
```

Konvertierungen

```
>>> z11.to(yd)
```

```
# L1: m ---> yd
```

```
L1(3.2808 yd)
```

1 Einleitung und Übersicht

```
>>> z21.to(ha)                # L2: m2 ----> ha
L2(0.0004 ha)
>>> L1(3.5,km).to(m)          # L1: km ----> m
L1(3500.0 m)
>>> T1(1.5,y).to(d)           # T1: y ----> d
T1(547.5 d)
>>> M(2.3,t).to(Ztr)          # M: t ----> Ztr
M(46.0 Ztr)
>>> TT(200,F).to(K)           # TT: F ----> K
TT(366.4833 K)
```

1.4 Informationen zum Modul

Einen ersten Überblick über die Möglichkeiten des Moduls bekommen Sie durch

```
>>> globalInfo("H")
Modul
=====
units.py
```

Methoden und Eigenschaften für das Rechnen mit Einheiten:

Klassen:

```
+ L1    [Längenmaße]
+ L2    [Flächenmaße]
+ L3    [Volumenmaße]
+ M      [Gewichtsmaße]
+ N      [Hilfsklasse für linksseitige Skalare]
+ T1     [Zeitmaße]
+ U(T1)  [Uhrzeiten]
+ T2     [Zeitmaße hoch 2]
+ TT     [Temperaturen]
+ G      [zusammengesetzte Objekte]
+ B(G)   [Beschleunigung]
+ F1(G)  [Kraft]
+ P(G)   [Leistung]
+ V(G)   [Geschwindigkeit]
+ W(G)   [Kraft]
```

globale Methoden:

```
__format_e(n)                Formatiert eine Zahl in Scientific Notation
                              (intern).
__ueberschrift(text,z="-")    Dient zum Ausgeben von Überschriften bei der
                              Ausgabe (intern).
```

1 Einleitung und Übersicht

<code>_repr_au(s)</code>	Formatiert eine Zahl als String in Scientific Notation oder mit Rundung (intern).
<code>_str_au(s)</code>	Formatiert eine Zahl als String in Scientific Notation oder mit Rundung (intern).
<code>allUnits(l)</code>	Listet alle berücksichtigten Einheiten (für L1, L2, L3, M, T1, T2) auf.
<code>au(l)</code>	Alias für <code>allUnits(l)</code>
<code>alle(l,m=...)</code>	Listet alle Instanzen der Klassen (L1, L2, L3, M, N, T1, T2, TT, G, U, V, B, F1, P, W) auf.
<code>beispiel(e)</code>	Liefert Beispiele für die globalen Methoden des Moduls.
<code>clear(f)</code>	Löscht alle Instanzen einer Klasse.
<code>dok(f)</code>	Liefert Informationen zu f (Methode oder Klasse).
<code>getClassNames()</code>	Liefert die Namen aller Klassen im Modul als Liste.
<code>gcn()</code>	Alias für <code>getClassNames()</code>
<code>getFunctionNames()</code>	Liefert die Namen aller globalen Methoden im Modul als Liste.
<code>gfn()</code>	Alias für <code>getFunctionNames()</code>
<code>globalInfo(m)</code>	Gibt Informationen über Klassen, Methoden und Variablen/Eigenschaften im Modul aus.
<code>gi(m)</code>	Alias für <code>globalInfo(m)</code>
<code>setVar(e=float, r=int, s=bool, t=char)</code>	Ermöglicht das nachträgliche Setzen der globalen Variablen <code>eps</code> , <code>science</code> , <code>rndg</code> , <code>trennz</code> .
<code>sv(e=float, r=int, s=bool, t=char)</code>	Alias für <code>setVar(e=float, r=int, s=bool, t=char)</code>
<code>sign(s)</code>	Liefert das Vorzeichen eines numerischen Werts/einer Instanz.
<code>typ(o)</code>	Gibt den Typ von o aus.
<code>frequenz(wert)</code>	Generiert eine G-Instanz mit dem Wert <code>wert</code> [Hz].
<code>joule(wert)</code>	Generiert eine W-Instanz (Arbeit) mit dem Wert <code>wert</code> [J].
<code>kelvin(wert)</code>	Generiert eine TT-Instanz (Temperatur) mit dem Wert <code>wert</code> [K].
<code>kilogramm(wert)</code>	Generiert eine M-Instanz (Gewicht) mit dem Wert <code>wert</code> [kg].
<code>meter(wert)</code>	Generiert eine L1-Instanz (Länge) mit dem Wert <code>wert</code> [m].
<code>meterS(wert)</code>	Generiert eine V-Instanz (Geschwindigkeit) mit dem Wert <code>wert</code> [m/s].
<code>mS(wert)</code>	Alias für <code>meterS(wert)</code>
<code>meterS2(wert)</code>	Generiert eine B-Instanz (Beschleunigung) mit dem Wert <code>wert</code> [m/s ²].
<code>mS2(wert)</code>	Alias für <code>meterS2(wert)</code>
<code>meter2(wert)</code>	Generiert eine L2-Instanz (Fläche) mit dem Wert <code>wert</code> [m ²].

1 Einleitung und Übersicht

<code>meter3(wert)</code>	Generiert eine L3-Instanz (Volumen) mit dem Wert <code>wert [m3]</code> .
<code>newton(wert)</code>	Generiert eine F1-Instanz (Kraft) mit dem Wert <code>wert [N]</code> .
<code>PS(wert)</code>	Generiert eine P-Instanz (Leistung) mit dem Wert <code>wert [PS]</code> .
<code>sekunde(wert)</code>	Generiert eine T1-Instanz (Temperatur) mit dem Wert <code>wert [s]</code> .
<code>sekunde2(wert)</code>	Generiert eine T2-Instanz mit dem Wert <code>wert [s2]</code> .
<code>watt(wert)</code>	Generiert eine P-Instanz (Leistung) mit dem Wert <code>wert [W]</code> .

Aliase:

<code>au</code>	Alias für <code>allUnits</code>
<code>gcn</code>	Alias für <code>getClassNames</code>
<code>gfn</code>	Alias für <code>getFunctionNames</code>
<code>gi</code>	Alias für <code>globalInfo</code>
<code>mS</code>	Alias für <code>meterS</code> (Geschwindigkeit)
<code>mS2</code>	Alias für <code>meterS2</code> (Beschleunigung)
<code>sv</code>	Alias für <code>setVar</code>

Anwendungen:

<code>druck(k, f)</code>	Berechnet Druck (=Kraft/Fläche).
<code>gprocm3(m, vol)</code>	Berechnet <code>g/cm3</code> .
<code>impuls(m, v)</code>	Berechnet Impuls (=Masse*Geschwindigkeit).
<code>literpro100km(vol, s)</code>	Berechnet liter/(100km).
<code>literprom2(l, f)</code>	Berechnet liter/m2.
<code>longweightsToM(l)</code>	Erzeugt aus der Liste [<code>lton</code> , <code>cwt</code> , <code>qu</code> , <code>stone</code> , <code>lb</code> , <code>oz</code> , <code>dr</code> , <code>grain</code>] eine M-Instanz (Gewicht).
<code>mpros(s, t)</code>	Berechnet m/s.
<code>mgprodl(m, vol)</code>	Berechnet mg/dl.
<code>myfiToL1(l)</code>	Erzeugt aus der Liste [<code>mi</code> , <code>yd</code> , <code>ft</code> , <code>inch</code>] eine L1-Instanz (Länge).
<code>Upromin(anz, t)</code>	Berechnet Umdrehung/min.
<code>ymdToT1(l)</code>	Erzeugt aus der Liste [<code>zy</code> , <code>zmo</code> , <code>zd</code> , <code>zh</code> , <code>zmi</code> , <code>zs</code>] eine T1-Instanz (Zeit).

In der Ausgabe finden Sie u. a. die Namen der verfügbaren Klassen und globalen Methoden (siehe auch Abschnitt auf der nächsten Seite). Welche weiteren Möglichkeiten die Methode `globalInfo` bietet, zeigt

```
>>> dok("globalInfo")
```

```
globalInfo
```

```
    Gibt Informationen über Klassen, Methoden und Eigenschaften/variablen im
```

Modul aus.

Aufruf: `globalInfo(m)` oder `gi(m)`

Zulässige Angaben für `m`:

"A" (alles) [Voreinstellung]

"V" (Variable)

"E" (Eigenschaften; wie "V")

"v" (Version)

"M" (Methoden)

"K" (Klassen)

"H" (Kopf)

Globale Methoden

In der obigen Auflistung auf der vorherigen Seite sind alle im Modul verfügbaren globalen Methoden aufgeführt. Kurzbeschreibungen aller globalen Methoden erhalten Sie durch `globalInfo("M")`, die einer bestimmten Methode durch die globale Methode `dok(name)`, beispielsweise

```
>>> dok("kilogramm")
```

```
kilogramm
```

```
Generiert eine M-Instanz (Gewicht) mit dem Wert wert [kg].
```

```
Aufruf: kilogramm(wert)
```

```
Parameter:
```

```
- wert (N, float oder int);
```

```
    Voreinstellung: 1
```

```
- n (Zeichenkette);
```

```
    Name der Instanz;
```

```
    Voreinstellung: leere Zeichenkette
```

```
mögliche Fehlermeldung:
```

```
- Operand hat keinen zulässigen Typ
```

- globale Methoden zum Generieren von Instanzen (Tabelle [1.1 auf der nächsten Seite](#)); vereinzelt Beispiele in den Darstellungen der einzelnen Klassen (insbesondere in den Abschnitten »Maßeinheiten und Instanzen«)
- globale Hilfsmethoden (Tabelle [1.2 auf Seite 11](#))
- globale Methoden, in denen die Möglichkeiten von Klassen kombiniert werden (Tabelle [1.3 auf Seite 13](#)); Beispiele dazu im Abschnitt [17.2 auf Seite 157](#)
- Kurzbezeichner (Aliase) für einige Methoden (Tabelle [1.4 auf Seite 14](#))

Tabelle 1.1: Globale Methoden zum Generieren von Instanzen

name	Bedeutung
<code>joule(wert)</code>	<p>erzeugt eine W-Instanz (Arbeit) mit dem Wert <i>wert</i> [J]; Beispiele →</p> <ul style="list-style-type: none"> □ Abschnitt »Instanzen« auf Seite 138 □ Abschnitt »Beispiele – Vergleiche« auf Seite 140 □ Abschnitt »Beispiele – logische Ausdrücke« auf Seite 141 □ Abschnitt »Beispiele – arithmetische Operatoren« auf Seite 141 □ Abschnitt »Beispiele – Konvertierung« auf Seite 143
<code>kelvin(wert)</code>	<p>erzeugt eine TT-Instanz mit dem Wert <i>wert</i> [K]; Beispiele → Abschnitt »Instanzen« auf Seite 76 und Abschnitt »Beispiele – Konvertierung« auf Seite 80</p>
<code>kilogramm(wert)</code>	<p>erzeugt eine M-Instanz mit dem Wert <i>wert</i> [kg]; Beispiele → Abschnitt »Instanzen« auf Seite 67 und Abschnitt »Beispiele – Konvertierung« auf Seite 72</p>
<code>meter(wert)</code>	<p>erzeugt eine L1-Instanz mit dem Wert <i>wert</i> [m]; Beispiele → Abschnitt »Instanzen« auf Seite 19 und Abschnitt »Beispiele – Konvertierung« auf Seite 24</p>
<code>meterS(wert)</code>	<p>erzeugt eine V-Instanz (Geschwindigkeit) mit dem Wert <i>wert</i> [m/s]; Beispiele →</p> <ul style="list-style-type: none"> □ Abschnitt »Instanzen« auf Seite 83 □ Abschnitt »Instanzen« auf Seite 108 □ Abschnitt »Beispiele – Vergleiche« auf Seite 111 □ Abschnitt »Beispiele – logische Ausdrücke« auf Seite 111 □ Abschnitt »Beispiele – Konvertierung« auf Seite 114
<code>mS(wert)</code>	Alias für <code>meterS(wert)</code>
<code>meterS2(wert)</code>	<p>erzeugt eine B-Instanz (Beschleunigung) mit dem Wert <i>wert</i> [m/s²]; Beispiele →</p> <ul style="list-style-type: none"> □ Abschnitt »Instanzen« auf Seite 118 □ Abschnitt »Beispiele – Vergleiche« auf Seite 120 □ Abschnitt »Beispiele – logische Ausdrücke« auf Seite 121 □ Abschnitt »Beispiele – arithmetische Operatoren« auf Seite 121 □ Abschnitt »Beispiele – Konvertierung« auf Seite 123
<code>mS2(wert)</code>	Alias für <code>meterS2(wert)</code>
<code>meter2(wert)</code>	<p>erzeugt eine L2-Instanz mit dem Wert <i>wert</i> [m²]; Beispiele → Abschnitt »Instanzen« auf Seite 29 und Abschnitt »Beispiele – Konvertierung« auf Seite 34</p>
<code>meter3(wert)</code>	<p>erzeugt eine L3-Instanz mit dem Wert <i>wert</i> [m³]; Beispiele → Abschnitt »Instanzen« auf Seite 39 und Abschnitt »Beispiele – Konvertierung« auf Seite 44</p>

(Fortsetzung auf nächster Seite)

(Fortsetzung)

newton (<i>wert</i>)	erzeugt eine F1-Instanz (Kraft) mit dem Wert <i>wert</i> [N]; Beispiele → <input type="checkbox"/> Abschnitt »Instanzen« auf Seite 128 <input type="checkbox"/> Abschnitt »Beispiele – Vergleiche« auf Seite 130 <input type="checkbox"/> Abschnitt »Beispiele – logische Ausdrücke« auf Seite 131 <input type="checkbox"/> Abschnitt »Beispiele – Konvertierung« auf Seite 133 <input type="checkbox"/> Abschnitt »Instanzen« auf Seite 138
sekunde (<i>wert</i>)	erzeugt eine T1-Instanz mit dem Wert <i>wert</i> [s]; Beispiele → Abschnitt »Instanzen« auf Seite 48 und Abschnitt »Beispiele – Konvertierung« auf Seite 53
sekunde2 (<i>wert</i>)	erzeugt eine T2-Instanz mit dem Wert <i>wert</i> [s2]; Beispiele → Abschnitt »Instanzen« auf Seite 57 und Abschnitt »Beispiele – Konvertierung« auf Seite 62
PS (<i>wert</i>)	generiert eine P-Instanz (Leistung) mit dem Wert <i>wert</i> [PS]; Beispiel → Abschnitt »Instanzen« auf Seite 148
watt (<i>wert</i>)	generiert eine P-Instanz (Leistung) mit dem Wert <i>wert</i> [watt]; Beispiel → <input type="checkbox"/> Abschnitt »Instanzen« auf Seite 148 <input type="checkbox"/> Abschnitt »Beispiele – Vergleiche« auf Seite 151 <input type="checkbox"/> Abschnitt »Beispiele – logische Ausdrücke« auf Seite 151 <input type="checkbox"/> Abschnitt »Beispiele – arithmetische Operatoren« auf Seite 151 <input type="checkbox"/> Abschnitt »Beispiele – Konvertierung« auf Seite 154

Tabelle 1.2: Globale Hilfsmethoden

Name	Bedeutung
allUnits (<i>klasse</i>)	Gibt die Zahl der Einheiten aus und listet alle berücksichtigten Einheiten (für L1, L2, L3, M, T1, T2) auf. Beispiele finden Sie im Abschnitt »Maßeinheiten und Instanzen« der betreffenden Klassenbeschreibungen.
au (<i>klasse</i>)	Alias für allUnits (<i>klasse</i>)
alle (<i>klasse</i> , <i>m=modus</i>)	Gibt die Zahl der Instanzen aus und listet alle Instanzen der Klassen (L1, L2, L3, M, N, T1, T2, TT, G, U, V, B, F1, P, W) auf. Beispiele finden Sie im Abschnitt »Maßeinheiten und Instanzen« der betreffenden Klassenbeschreibungen.
beispiel (<i>bezeichnung</i>)	Liefert Beispiele für die globale Methode <i>bezeichnung</i> . (Fortsetzung auf nächster Seite)

(Fortsetzung)

Name	Bedeutung
<code>clear(klasse)</code>	Löscht alle Instanzen der Klasse <i>klasse</i> . Zusätzlich wird die Zahl der gelöschten Instanzen ausgegeben. Ein Beispiel finden Sie auf dieser Seite.
<code>dok(name)</code>	Liefert Informationen zu <i>name</i> (Methode oder Klasse). Beispiele finden Sie im Abschnitt »Übersicht« der betreffenden Klassenbeschreibungen.
<code>getClassNames()</code>	Liefert die Namen aller Klassen im Modul als Liste. Ein Beispiel finden Sie auf dieser Seite.
<code>gcn()</code>	Alias für <code>getClassNames()</code>
<code>getFunctionNames()</code>	Liefert die Namen aller globalen Methoden im Modul als Liste. Ein Beispiel finden Sie auf dieser Seite.
<code>gfn()</code>	Alias für <code>getFunctionNames()</code>
<code>globalInfo(name)</code>	Gibt Informationen über Klassen, Methoden und Eigenschaften des Moduls aus. Beispiele finden Sie im aktuellen Abschnitt »Informationen zum Modul«.
<code>gi(name)</code>	Alias für <code>globalInfo(name)</code>
<code>setVar(e=wert,r=zahl,s=ein/aus,t=zeichen)</code>	Ermöglicht das nachträgliche Setzen der globalen Eigenschaften <code>eps</code> , <code>science</code> , <code>rndg</code> , <code>trennz</code> . Ein Beispiel finden Sie im Abschnitt »Globale Eigenschaften« auf Seite 14.
<code>sv(e=wert,r=zahl,s=ein/aus,t=zeichen)</code>	Alias für <code>setVar(...)</code>
<code>sign(wert)</code>	Liefert das Vorzeichen des numerischen Werts <i>wert</i> bzw. der Instanz <i>wert</i> . Beispiele finden Sie auf dieser Seite.
<code>typ(instanz)</code>	Gibt den Typ von <i>instanz</i> aus. Beispiele finden Sie auf dieser Seite.

Einige Beispiele

```

>>> ll = getClassNames()           # liefert Liste
>>> ll[0]; ll[len(ll) - 1]         # 1. und letztes Element:
'B'
'W'
>>> lll = getFunctionNames()       # liefert Liste
>>> lll[1]; lll[len(lll) - 2]      # 2. und vorletztes Element:
'Upromin'
'watt'
>>> sign(z11);sign(2.5);sign(-meter(3)) # Vorzeichen:
1

```

```

1
-1
>>> typ(1.1); typ(2.3e10); typ(watt(1)) # Typ
'float'
'float'
'p'
>>> alle("TT") # alle TT-Instanzen
3 Element(e):

Name      : TT(300 K)
Art       : TT-Instanz (Temperatur)
Wert      : 300 (in K)
Einheit   : K [Temperaturmaß: Kelvin]
interner Wert: 300 (in K)
-----

Name      : TT(200 F)
Art       : TT-Instanz (Temperatur)
Wert      : 200 (in F)
Einheit   : F [Temperaturmaß: Fahrenheit]
interner Wert: 366.4833 (in K)
-----

Name      : TT(200 F)--->K
Art       : TT-Instanz (Temperatur)
Wert      : 366.4833 (in K)
Einheit   : K [Temperaturmaß: Kelvin]
interner Wert: 366.4833 (in K)
-----

>>> clear("A") # alle Instanzen löschen
146

```

Tabelle 1.3: Globale Methoden, in denen Klassen kombiniert werden

Name	Bedeutung
druck (kraft, fläche)	Berechnet Druck (= kraft/fläche).
frequenz (wert)	Berechnet Frequenz (=wert/s) [Hz]
gproc3 (masse, volumen)	Berechnet g/cm3 aus masse/volumen.
impuls (masse, geschwindigkeit)	Berechnet Impuls (= masse * geschwindigkeit).
literpro100km (volumen, strecke)	Berechnet liter/hkm.
literprom2 (volumen, fläche)	Berechnet liter/m2 aus volumen/fläche.
mpros (strecke, zeit)	Berechnet m/s aus strecke/zeit (Geschwindigkeit).
mgprod1 (masse, volumen)	Berechnet mg/dl aus masse/volumen.
Upromin (anzahl, zeit)	Berechnet umdrehung/minute.

Beispiele dazu finden Sie in Abschnitt 17.2 auf Seite 157.

Tabelle 1.4: Kurzbezeichner (Aliase) für einige Methoden und Klassen

Name	Bedeutung
<code>au(parameter)</code>	Alias für die globale Methode <code>allUnits(parameter)</code>
<code>gcn(parameter)</code>	Alias für die globale Methode <code>getClassNames(parameter)</code>
<code>gfn(parameter)</code>	Alias für die globale Methode <code>getFunctionNames(parameter)</code>
<code>gi(parameter)</code>	Alias für die globale Methode <code>globalInfo(parameter)</code>
<code>L</code>	Alias für <code>L1</code>
<code>mS(parameter)</code>	Alias für die globale Methode <code>meterS(parameter)</code> (Geschwindigkeit)
<code>mS2(parameter)</code>	Alias für die globale Methode <code>meterS2(parameter)</code> (Beschleunigung)
<code>sv(parameter)</code>	Alias für die globale Methode <code>setVar(parameter)</code>
<code>T</code>	Alias für <code>T1</code>

Globale Eigenschaften

Einen Überblick über die globalen Eigenschaften und ihre aktuellen Werte erhalten Sie durch `globalInfo('E')`; mittels `setVar(parameter)` können Sie einige Werte neu setzen.

```
>>> globalInfo("E")      # Informationen zu globalen Eigenschaften
```

Globale Variablen/Eigenschaften

```
eps      1e-16      für Vergleiche auf Gleichheit: ganz klein
rndg     4          für Rundung: Nachkommastellen
science  False      Darstellung von Instanzen in Science Notation
trennz                   Trennzeichen zw. Maß und Maßeinheit in __str__/__repr__
```

Diese Eigenschaften können mit Hilfe der globalen Methode `setVar` geändert werden.

Außerdem:

```
modul    units.py   Name des Moduls
version  9.16.2     aktuelle Version des Moduls
date     2018-07-29 Datum der letzten Änderung
```

```
>>> dok("setVar")      # Information zur globalen Methode setVar
```

`setVar`

Ermöglicht das nachträgliche Setzen der globalen Eigenschaften (e)ps, (r)ndg, (s)cience, (t)rennz.

Aufruf: `setVar(e=float, r=int, s=bool, t=char)` oder `sv(e=float, r=int, s=bool, t=char)`

(e)ps : Genauigkeit von float-Zahlen

(r)ndg, : Nachkommazahlen

1 Einleitung und Übersicht

```
(s)cience : wiss. Notation/Fixkomma
(t)rennz   : Trennzeichen zwischen Maßzahl und Maßeinheit
keine Voreinstellungen
eingestellte Werte können durch gi("V") aufgelistet werden
>>> setVar(r=2, t=",") # globale Eigenschaften setzen
```

Mit der letzten Anweisung wird für die Instanz-Darstellungen

- die Zahl der signifikanten Nachkommastellen mit 2 und
- das Zeichen »«,« als Trennzeichen zwischen Maßzahl und Maßeinheit

festgelegt.

Durch einen Aufruf der globalen Methode `globalInfo('V')` (oder `globalInfo('E')`) können übrigens die neuen aktuellen Werte aufgelistet werden:

```
>>> globalInfo("V")
Globale Variablen/Eigenschaften
-----
eps      1e-16      für Vergleiche auf Gleichheit: ganz klein
rndg     2          für Rundung: Nachkommastellen
science False      Darstellung von Instanzen in Science Notation
trennz   ,         Trennzeichen zw. Maß und Maßeinheit in __str__/__repr__
```

Diese Eigenschaften können mit Hilfe der globalen Methode `setVar` geändert werden.

Außerdem:

```
modul    units.py    Name des Moduls
version  9.16.2      aktuelle Version des Moduls
date     2018-07-29  Datum der letzten Änderung
```

2 Basisklasse L1

L1 realisiert das Rechnen mit Längenmaßen.

2.1 Übersicht

Globale Informationen über die Klasse L1

Eine Übersicht über die Klasse L1 erhalten Sie durch die globale Methode `dok("L1")` oder – etwas ausführlicher – mit der L1-Methode `L1.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("L1.classInfo")`:

```
>>> dok("L1.classInfo")
L1.classInfo
    Gibt Informationen zur Klasse L1 und ihre Methoden aus.

    Aufruf: L1.classInfo(m=art) oder L1.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "M"/Methoden   : Methoden
    + "V"/Variablen  : Variablen/Eigenschaften
    + "E"/Einheiten  : Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> L1.classInfo("H")
Class L1

    Realisiert das Rechnen mit L1-Instanzen (Längen).

    Parameter:

    value=1: Längenwert;           Voreinstellung: 1
    unit=m : L1-Maßeinheit;       Voreinstellung: m
    n=""   : Name der L1-Instanz; Voreinstellung: leere Zeichenkette

    Methoden:

    l : T1-Instanzen (Länge)
    o : Instanz
    i : Integer
```

2 Basisklasse L1

u : eine zulässige L1-Maßeinheit

<code>l.__abs__()</code> oder <code>abs(l)</code>	Liefert den Absolut-Betrag der L1-Instanz.
<code>l.__add__(l)</code> oder <code>l + l</code>	Realisiert die Addition zweier L1-Instanzen.
<code>l.__eq__(l)</code> oder <code>l == l</code>	Realisiert den Vergleich <code>==</code> in L1.
<code>l.__ge__(l)</code> oder <code>l >= l</code>	Realisiert den Vergleich <code>>=</code> in L1.
<code>l.__gt__(l)</code> oder <code>l > l</code>	Realisiert den Vergleich <code>></code> in L1.
<code>L1.__init__(value=1, unit=m, n="")</code>	Initialisiert eine L1-Instanz.
<code>l.__le__(l)</code> oder <code>l <= l</code>	Realisiert den Vergleich <code><=</code> in L1.
<code>l.__lt__(l)</code> oder <code>l < l</code>	Realisiert den Vergleich <code><</code> in L1.
<code>l.__mul__(o)</code> oder <code>l * o</code>	Realisiert die Multiplikation in L1.
<code>l.__ne__(l)</code> oder <code>l != l</code>	Realisiert den Vergleich <code>!=</code> in L1.
<code>l.__neg__()</code> oder <code>-(l)</code>	Realisiert negatives Vorzeichen in L1.
<code>l.__pos__()</code> oder <code>+(l)</code>	Realisiert positives Vorzeichen in L1.
<code>l.__pow__(i)</code> oder <code>l ** i</code>	Realisiert das Potenzieren in L1.
<code>l.__repr__()</code> oder <code>repr(l)</code>	Repräsentiert eine L1-Instanz.
<code>l.__str__()</code> oder <code>str(l)</code>	Repräsentiert eine L1-Instanz.
<code>l.__sub__(l)</code> oder <code>l - l</code>	Realisiert die Subtraktion zweier L1-Instanzen.
<code>l.__truediv__(o)</code> oder <code>l / o</code>	Realisiert die Division in L1.
<code>L1.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse L1.
<code>L1.classInfo(m=art)</code>	Gibt Informationen zur Klasse L1 und ihre Methoden aus.
<code>L1.ci(m=art)</code>	Alias für <code>L1.classInfo(m=art)</code>
<code>L1.description()</code>	Gibt eine Kurzbeschreibung der Klasse L1 aus.
<code>l.info(modus)</code>	Gibt Informationen über eine L1-Instanz aus.
<code>l.to(u)</code>	Realisiert die Konvertierung einer L1-Instanz in eine andere Maßeinheit.

Weitere Informationen

- Methoden der Klasse → Abschnitt 2.2
- Maßeinheiten und Instanzen → Abschnitt 2.3 auf der nächsten Seite
- Eigenschaften der Instanzen → Abschnitt 2.5 auf Seite 23
- Zulässige Operationen → Abschnitt 2.4 auf Seite 22
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 2.6 auf Seite 24

2.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("L1")` auf der vorherigen Seite liefert u. a. eine Auflistung der L1-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 2.1 auf der nächsten Seite aufgeführt.

Tabelle 2.1: Methoden der Klasse L1

Name	Bedeutung
<code>L1.beispiel(<i>e</i>)</code>	Liefert Beispiele für die Methoden der Klasse L1.
<code>L1.classInfo(<i>art</i>)</code>	Gibt Informationen zur Klasse L1 und ihre Methoden aus.
<code>L1.ci(<i>art</i>)</code>	Alias für <code>classInfo(<i>art</i>)</code>
<code>L1.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse L1 aus.
<code><i>l</i>.info(<i>modus</i>)</code>	Gibt Informationen über die L1-Instanz <i>l</i> aus.
<code><i>l</i>.to(<i>u</i>)</code>	Realisiert die Konvertierung der L1-Instanz <i>l</i> in eine andere Maßeinheit <i>u</i> .

Ausführlichere Informationen über eine bestimmte L1-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `L1.classInfo("M")` für alle L1-Methoden.

Beispiele für dok

```
>>> dok("L1.to")
L1.to
    Realisiert die Konvertierung einer L1-Instanz in eine andere Maßeinheit.

    Aufruf: l.to(u)
    erlaubter Typ des Operanden:
    - eine der zulässigen L1-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("L1.info")
L1.info
    Gibt Informationen über eine L1-Instanz aus.

    Aufruf: l.info(m)
    m Modus
        kurz nur Grund-Eigenschaften
            [Voreinstellung]
        lang auch Darstellung in anderen Einheiten
```

2.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die Maßeinheiten in L1 liefern die globalen Methoden `allUnits("L1")` und `au("L1")`:

```
>>> allUnits("L1")
Längenmaße (in m): 30 Einträge
    aa      0.0      Längenmaß: Ångström
```


2 Basisklasse L1

am	0.0	Längenmaß: atto meter
bp	0.0	Längenmaß im Druckergewerbe: big point; 72 bp/in
cc	0.0	Längenmaß im Druckergewerbe: cicero; 1 cc = 12 dd
cm	0.01	Längenmaß: Zentimeter (centi meter)
dd	0.0	Längenmaß im Druckergewerbe: Didot; 1157 dd = 1238 pt
dm	0.1	Längenmaß: Dezimeter (deci meter)
fm	0.0	Längenmaß: femto meter
ft	0.3	amerik. Längenmaß: Fuß (feet)
Gm	1000000000.0	Längenmaß: Giga meter
hkm	100000.0	Längenmaß: 100 kilometer
inch	0.03	amerik. Längenmaß: Zoll (inch)
km	1000.0	Längenmaß: kilometer (kilo meter)
ly	9460730472580800.0	Längenmaß: Lichtjahr (light year)
m	1	Längenmaß: Meter (meter)
mi	1609.34	amerik. Längenmaß: Meile (mile)
mm	0.0	Längenmaß: Millimeter (milli meter)
Mm	1000000.0	Längenmaß: Mega meter
my	0.0	Längenmaß: mikro meter
nm	0.0	Längenmaß: nano meter
pc	0.0	Längenmaß im Druckergewerbe: Pica; 12 pt/pc
pm	0.0	Längenmaß: pico meter
Pm	1000000000000000.0	Längenmaß: Peta meter
pt	0.0	Längenmaß im Druckergewerbe: Point (point)
sm	1852	amerik. Längenmaß: Seemeile
sp	0.0	Längenmaß im Druckergewerbe: scaled point; 65536 sp/pt
Tm	1000000000000.0	Längenmaß: Tera meter
yd	0.91	amerik. Längenmaß: yard
ze	0.0	Längenmaß im Druckergewerbe: Zeiligkeit
Zoll	0.03	amerik. Längenmaß: Alias für Inch

Interessant sind die Maßeinheiten für das Druckergewerbe (bp, cc, dd, pc, pt, sp, ze) und die amerikanischen Maßeinheiten (ft, inch, mi, sm, yd, Zoll).

Instanzen

Eigene L1-Instanzen können Sie mittels der Anweisung `L1(länge, einheit, n=name)` oder der globalen Methoden `meter(wert)` (Längenmaß) bzw. `myfiToL1(liste)` (Längenmaß, amerikanisch) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

Beispiele

```
>>> l11 = L1(2, m)
>>> l12 = L1(-2, mm)
>>> l13 = -L1(0.5, km)
```

```

>>> l14 = +meter(3.75)  # globale Methode
>>> l    = [1, 2, 3, 4]  # [mi, yd, ft, inch]
>>> l15 = myfiToL1(l)    # globale Methode; [mi, yd, ft, inch] --> L1
>>> l11
L1(2,m)
>>> l12
L1(-2,mm)
>>> l13
L1(-0.5,km)
>>> l14
L1(3.75,m)
>>> l15
L1(1612.19,m)

```

Mögliche Fehlerquellen

Der Versuch, eine eigene L1-Instanz zu generieren, kann misslingen:

- Unit ist hier nicht bekannt/zulässig: Nur die bei einem Aufruf der globalen Methode `allUnits("L1")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("L1.__init__")`

```

>>> dok("L1.__init__")
L1.__init__
    Initialisiert eine L1-Instanz.

    Aufruf: L1.__init__(value=1, unit=m, n="")
    Parameter:
    - value: Wert der L1-Instanz;
              Voreinstellung: 1
    - unit : eine zulässige L1-Maßeinheit;
              Voreinstellung: m
    - n    : Name der L1-Instanz;
              Voreinstellung: leere Zeichenkette
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig

```

Informationen über Instanzen

Informationen über eine einzelne L1-Instanz bekommen Sie mit der L1-Methode `info`, beispielsweise für l11:

```

>>> l11.info()
Name       : L1(2,m)
Art        : L1-Instanz (Länge)
Wert       : 2 (in m)
Einheit    : m [Längenmaß: Meter (meter)]

```

```
interner Wert: 2 (in m)
```

Informationen über alle L1-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("L1")
```

```
7 Element(e):
```

```
Name      : L1(2,m)
Art       : L1-Instanz (Länge)
Wert      : 2 (in m)
Einheit   : m [Längenmaß: Meter (meter)]
interner Wert: 2 (in m)
```

```
Name      : L1(-2,mm)
Art       : L1-Instanz (Länge)
Wert      : -2 (in mm)
Einheit   : mm [Längenmaß: Millimeter (milli meter)]
interner Wert: -0.0 (in m)
```

```
Name      : L1(0.5,km)
Art       : L1-Instanz (Länge)
Wert      : 0.5 (in km)
Einheit   : km [Längenmaß: kilometer (kilo meter)]
interner Wert: 500.0 (in m)
```

```
Name      : L1(-0.5,km)
Art       : L1-Instanz (Länge)
Wert      : -0.5 (in km)
Einheit   : km [Längenmaß: kilometer (kilo meter)]
interner Wert: -500.0 (in m)
```

```
Name      : meter(3.75)
Art       : L1-Instanz (Länge)
Wert      : 3.75 (in m)
Einheit   : m [Längenmaß: Meter (meter)]
interner Wert: 3.75 (in m)
```

```
Name      : L1(3.75,m)
Art       : L1-Instanz (Länge)
Wert      : 3.75 (in m)
Einheit   : m [Längenmaß: Meter (meter)]
```

```

interner Wert: 3.75 (in m)
-----
Name          : L1(1612.19,m)
Art           : L1-Instanz (Länge)
Wert          : 1612.19 (in m)
Einheit       : m [Längenmaß: Meter (meter)]
interner Wert: 1612.19 (in m)
-----

```

2.4 Operatoren

Vergleiche

Instanzen der Klasse L1 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> l11, l12, l13, l14
(L1(2,m), L1(-2,mm), L1(-0.5,km), L1(3.75,m))
>>> l11 <= l12
False
>>> l13 >= l14
False
>>> l12 != L1(23, m)
True
>>> l14 == L1(3.4, cm)
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> l11
L1(2,m)
>>> (L1(1, m) <= l11) and (l11 <= L1(5, m))      # l11 zwischen 1m und 5m
True

```

```
>>> not ((l11 == L1(1, m)) or (l11 == L1(5, m))) # l11 weder 1m noch 5m
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse L1 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 2.2

Tabelle 2.2: Klasse L1, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	L2	L3	G	G	L1	G	G	G	G	G	G	L1
/	T2	G	G	Skalar	G	G	G	L1	G	V	B	G	T1	G	L1
+	—	—	—	—	—	—	—	L1	—	—	—	—	—	—	—
-	—	—	—	—	—	—	—	L1	—	—	—	—	—	—	—
**	—	—	—	—	—	—	—	?	—	—	—	—	—	—	?

Beispiele – arithmetische Operatoren

```
>>> l11, l12, l13, l14
(L1(2,m), L1(-2,mm), L1(-0.5,km), L1(3.75,m))
>>> l11 + l12
L1(2.0,m)
>>> l13 - L1(23, m)
L1(-523.0,m)
>>> l14 * L1(3.4, mm)
L2(0.01,m2)
>>> l11 / l12
-1000.0
>>> l13 ** 3
L3(-125000000.0,m3)
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit L1-Instanzen erhalten Sie durch `L1.beispiel(beispiel)`. Welche Angaben für *beispiel* möglich sind, erfahren Sie durch die globale Methode `dok("L1.beispiel")`, beispielsweise `L1.beispiel()` für alle Beispiele in L1.

2.5 Eigenschaften der L1-Instanzen

Jede L1-Instanz (jedes L1-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der L1-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch

gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `L1.classInfo("V")`. Danach stehen in der Klasse L1 die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> L1.classInfo("V")
```

Eigenschaften der L1-Instanzen:

```
l.v      Wert der Instanz (gemessen in l.u)
l.u      Maßeinheit der Instanz (L1-Maßeinheit)
l.name   Name der Instanz
l.internal interner Wert
```

Beispiele

```
>>> l13
```

```
L1(-0.5,km)
```

```
>>> l13.info()
```

```
Name      : L1(-0.5,km)
Art       : L1-Instanz (Länge)
Wert      : -0.5 (in km)
Einheit   : km [Längenmaß: kilometer (kilo meter)]
interner Wert: -500.0 (in m)
```

```
-----
```

```
>>> l13.v; l13.u; l13.name; l13.internal
```

```
-0.5
```

```
'km'
```

```
'L1(-0.5,km)'
```

```
-500.0
```

2.6 Konvertierung

Instanzen der Klasse L1 können mit der Hilfe der L1-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden:

Beispiele – Konvertierung

```
>>> l12, l12, l13, l14
```

```
(L1(-2,mm), L1(-2,mm), L1(-0.5,km), L1(3.75,m))
```

```
>>> l11.to(km)      # m --> km
```

```
L1(0.0,km)
```

```
>>> l12.to(m)       # mm --> m
```

```
L1(-0.0,m)
```

```
>>> l13.to(cm)      # km --> cm
```

```
L1(-50000.0,cm)
```

```

>>> l14.to(mm)          # m --> mm
L1(3750.0,mm)
>>> L1(2,inch).to(cm)   # Zoll --> cm
L1(5.08,cm)
>>> L1(10,m).to(yd)     # m --> yd
L1(10.94,yd)
>>> L1(1.5,ze).to(mm)   # ze --> mm
L1(6.35,mm)
>>> meter(3.5).to(yd)   # m --> yd # globale Methode
L1(3.83,yd)
>>> l14.to(s)           # m --> s ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 4864, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: s

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

3 Basisklasse L2

L2 realisiert das Rechnen mit Flächenmaßen.

3.1 Übersicht

Globale Informationen über die Klasse L2

Eine Übersicht über die Klasse L2 erhalten Sie durch die globale Methode `dok("L2")` oder – etwas ausführlicher – mit der L2-Methode `L2.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("L2.classInfo")`:

```
>>> dok("L2.classInfo")
L2.classInfo
    Gibt Informationen zur Klasse L2 und ihren Methoden aus.

    Aufruf: L2.classInfo(m=art) oder L2.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
    + "E"/Einheiten  : Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> L2.classInfo("H")
Class L2

    Realisiert das Rechnen mit L2-Instanzen (Flächen).

    Aufruf: L2.__init__(value=1, unit=m2, n="")

    Parameter:

    value=1: Wert;
              Voreinstellung: 1
    unit=m2: L2-Maßeinheit;
              Voreinstellung: m2
    n=""    : Name der L2-Instanz;
              Voreinstellung: leere Zeichenkette
```


Methoden:

```
f : L2-Instanz (Fläche)
n : Zahl(Integer/Float)
o : Instanz
i : Integer
u : eine der zulässigen T2-Maßeinheiten

f.__abs__() oder abs(f)      Liefert den Absolut-Betrag der L2-Instanz.
f.__add__(f) oder f + f     Realisiert die Addition zweier L2-Instanzen.
f.__eq__(f) oder f == f     Realisiert den Vergleich ==.
f.__ge__(f) oder f >= f     Realisiert den Vergleich >=.
f.__gt__(f) oder f > f      Realisiert den Vergleich >.
L2.__init__(value=1,unit=m2,n="") Initialisiert eine L2-Instanz.
f.__le__(f) oder f <= f     Realisiert den Vergleich <=.
f.__lt__(f) oder f < f      Realisiert den Vergleich <.
f.__mul__(o) oder f * o     Realisiert die Multiplikation.
f.__neg__() oder -(f)       Realisiert negatives Vorzeichen für eine
                             L2-Instanz.
f.__pos__() oder +(f)       Realisiert positives Vorzeichen für eine
                             L2-Instanz.
f.__pow__(p) oder f ** p    Realisiert das Potenzieren einer L2-Instanz.
f.__repr__() oder repr(f)   Repräsentiert eine L2-Instanz.
f.__str__() oder str(f)     Repräsentiert eine L2-Instanz.
f.__sub__(f) oder f - f     Realisiert die Subtraktion zweier L2-Instanzen.
f.__truediv__(o) oder f / o Realisiert die Division einer L2-Instanz.
L2.beispiel(e)              Liefert Beispiele für die Methoden der Klasse L2.
L2.classInfo(m=art)         Gibt Informationen zur Klasse L2 und ihrer
                             Methoden aus.
L2.ci(m=art)                Alias für L2.classInfo(m=art)
L2.description()            Gibt eine Kurzbeschreibung der Klasse aus.
f.info(modus)               Gibt Informationen über eine L2-Instanz aus.
f.to(u)                     Realisiert die Konvertierung einer L2-Instanz in
                             eine andere Maßeinheit.
```

Weitere Informationen

- Methoden der Klasse → Abschnitt [3.2 auf der nächsten Seite](#)
- Maßeinheiten und Instanzen → Abschnitt [3.3 auf Seite 29](#)
- Eigenschaften der Instanzen → Abschnitt [3.5 auf Seite 33](#)
- Zulässige Operationen → Abschnitt [3.4 auf Seite 31](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [3.6 auf Seite 34](#)

3.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("L2")` auf Seite 26 liefert u. a. eine Auflistung der L2-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 3.1 aufgeführt.

Tabelle 3.1: Methoden der Klasse L2

Name	Bedeutung
<code>L2.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse L2.
<code>L2.classInfo(art)</code>	Gibt Informationen zur Klasse L2 und ihrer Methoden aus.
<code>L2.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>L2.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse aus.
<code>l.info(modus)</code>	Gibt Informationen über die L2-Instanz <i>l</i> aus.
<code>l.to(u)</code>	Realisiert die Konvertierung der L2-Instanz <i>l</i> in eine andere Maßeinheit <i>u</i> .

Ausführlichere Informationen über eine bestimmte L2-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `L2.classInfo("M")` für alle L2-Methoden.

Beispiele für dok

```
>>> dok("L2.to")
L2.to
    Realisiert die Konvertierung einer L2-Instanz in eine andere Maßeinheit.

    Aufruf: f.to(u)
    erlaubter Typ des Operanden:
    - eine der zulässigen L2-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("L2.info")
L2.info
    Gibt Informationen über eine L2-Instanz aus.

    Aufruf: f.info(m)
    m Modus
    kurz   nur Grund-Eigenschaften
           [Voreinstellung]
    lang   auch Darstellung in anderen Einheiten
```

3.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die L2-Maßeinheiten liefern die globalen Methoden `allUnits("L2")` und `au("L2")`:

```
>>> allUnits("L2")
Flächenmaße (in m2): 22 Einträge
aa2      0.0          Flächenmaß: (Aangström) hoch 2
am2      0.0          Flächenmaß: (atto meter) hoch 2
ar       100.0        Flächenmaß: ar
cm2      0.0          Flächenmaß: (centi meter) hoch 2
dm2      0.01         Flächenmaß: (deci meter) hoch 2
fm2      0.0          Flächenmaß: (femto meter) hoch 2
ft2      0.09         amerik. Flächenmaß: (feet) hoch 2
Gm2      1e+18        Flächenmaß: (Giga meter) hoch 2
ha       10000.0      Flächenmaß: hektar
inch2    0.0          amerik. Flächenmaß: (Zoll) hoch 2
km2      1000000.0    Flächenmaß: (kilo meter) hoch 2
m2       1            Flächenmaß: (meter) hoch 2
mi2      2589988.11   amerik. Flächenmaß: (mile) hoch 2
Mm2      1000000000000.0 Flächenmaß: (Mega meter) hoch 2
mm2      0.0          Flächenmaß: (milli meter) hoch 2
my2      0.0          Flächenmaß: (mikro meter) hoch 2
nm2      0.0          Flächenmaß: (nano meter) hoch 2
pm2      0.0          Flächenmaß: (pico meter) hoch 2
Pm2      1e+30        Flächenmaß: (Peta meter) hoch 2
sm2      3429904      amerik. Flächenmaß: (Seemeile) hoch 2
Tm2      1e+24        Flächenmaß: (Tera meter) hoch 2
yd2      0.84         amerik. Flächenmaß: (yard) hoch 2
-----
```

Interessant sind die amerikanischen Flächenmaße (`ft2`, `inch2`, `mi2`, `sm2`, `yd2`).

Instanzen

Eigene L2-Instanzen können Sie mittels der Anweisung `L2(fläche, einheit, n=name)` oder der globalen Methode `meter2(wert)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> l21 = L2(2, m2)
>>> l22 = L2(-2, mm2)
>>> l23 = -L2(0.5, km2)
>>> l24 = +meter2(3.75) # globale Methode
>>> l21
L2(2,m2)
>>> l22
```

```
L2(-2,mm2)
>>> 123
L2(-0.5,km2)
>>> 124
L2(3.75,m2)
```

Mögliche Fehlerquellen

Der Versuch, eine eigene L2-Instanz zu generieren, kann misslingen:

- **Unit ist hier nicht bekannt/zulässig:** Nur die bei einem Aufruf der globalen Methode `allUnits("L2")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("L2.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne L2-Instanz erhalten Sie mit der L2-Methode `info`, beispielsweise für 121:

```
>>> 121.info()
Name          : L2(2,m2)
Art           : L2-Instanz (Länge)
Wert          : 2 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 2 (in m2)
-----
```

Informationen über alle L2-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("L2")
7 Element(e):

Name          : L1(3.75,m) * (L1(3.4,mm))
Art           : L2-Instanz (Länge)
Wert          : 0.01 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 0.01 (in m2)
-----
```

```
Name          : L2(2,m2)
Art           : L2-Instanz (Länge)
Wert          : 2 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 2 (in m2)
-----
```

```
Name          : L2(-2,mm2)
Art           : L2-Instanz (Länge)
```

3 Basisklasse L2

```
Wert          : -2 (in mm2)
Einheit       : mm2 [Flächenmaß: (milli meter) hoch 2]
interner Wert: -0.0 (in m2)
-----
```

```
Name          : L2(0.5,km2)
Art           : L2-Instanz (Länge)
Wert          : 0.5 (in km2)
Einheit       : km2 [Flächenmaß: (kilo meter) hoch 2]
interner Wert: 500000.0 (in m2)
-----
```

```
Name          : L2(-0.5,km2)
Art           : L2-Instanz (Länge)
Wert          : -0.5 (in km2)
Einheit       : km2 [Flächenmaß: (kilo meter) hoch 2]
interner Wert: -500000.0 (in m2)
-----
```

```
Name          : meter2(3.75)
Art           : L2-Instanz (Länge)
Wert          : 3.75 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 3.75 (in m2)
-----
```

```
Name          : L2(3.75,m2)
Art           : L2-Instanz (Länge)
Wert          : 3.75 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 3.75 (in m2)
-----
```

3.4 Operatoren

Vergleiche

Instanzen der Klasse L2 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> 121, 122, 123, 124
(L2(2,m2), L2(-2,mm2), L2(-0.5,km2), L2(3.75,m2))
>>> 121 <= 122
False
>>> 123 >= 124
False
>>> 122 != L2(23, m2)
True
>>> 124 == L2(3.4, cm2)
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> 121
L2(2,m2)
>>> (L2(1, m2) <= 121) and (121 <= L2(5, m2))      # l21 zwischen 1m2 und 5m2
True
>>> not ((121 == L2(1, m2)) or (121 == L2(5, m2))) # l21 weder 1m2 noch 5m2
True

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse L2 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 3.2.

Tabelle 3.2: Klasse L2, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	L3	G	G	G	L2	G	G	G	G	G	G	L2
/	G	G	G	L1	Skalar	G	G	L2	G	G	G	G	G	G	L2
+	–	–	–	–	–	–	–	L3	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	L3	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```

>>> l12, l22, l23, l24
(L1(-2,mm), L2(-2,mm2), L2(-0.5,km2), L2(3.75,m2))
>>> l21 + l22
L2(2.0,m2)
>>> l23 - L2(23, m2)
L2(-500023.0,m2)
>>> l24 * L2(3.4, km2)
G(L2(3.75,m2) * (L2(3.4,km2)))
>>> l21 / l22
-1000000.0
>>> l23 ** 0.5
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 5170, in __pow__
    return L(math.sqrt(self.internal), _m, n=name)
ValueError: math domain error

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit L2-Instanzen erhalten Sie durch `L2.beispiel(beispiel)`. Welche Angaben für *beispiel* möglich sind, erfahren Sie durch die globale Methode `dok("L2.beispiel")`, beispielsweise `L2.beispiel()`.

3.5 Eigenschaften der L2-Instanzen

Jede L2-Instanz (jedes L2-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der L2-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `L2.classInfo("V")`. Danach stehen in der Klasse L2 die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> L2.classInfo("V")
Eigenschaften der L2-Instanzen:

l.v          Wert der Instanz (gemessen in l.u)
l.u          Maßeinheit der Instanz (L2-Maßeinheit)
l.name       Name der Instanz
l.internal   interner Wert

```

Beispiele

```
>>> l24
L2(3.75,m2)
>>> l24.info()
Name          : L2(3.75,m2)
Art           : L2-Instanz (Länge)
Wert          : 3.75 (in m2)
Einheit       : m2 [Flächenmaß: (meter) hoch 2]
interner Wert: 3.75 (in m2)
-----

>>> l24.v
3.75
>>> l24.u
'm2'
>>> l24.name
'L2(3.75,m2)'
>>> l24.internal
3.75
```

3.6 Konvertierung

Instanzen der Klasse L2 können mit der Hilfe der L2-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden:

Beispiele – Konvertierung

```
>>> l21, l22, l23, l24
(L2(2,m2), L2(-2,mm2), L2(-0.5,km2), L2(3.75,m2))
>>> l21.to(cm2)          # m2 --> cm2
L2(20000.0,cm2)
>>> l22.to(cm2)          # mm2 --> cm2
L2(-0.02,cm2)
>>> l23.to(m2)           # km2 --> m2
L2(-500000.0,m2)
>>> l24.to(km2)          # m2 --> km2
L2(0.0,km2)
>>> L2(2,inch2).to(cm2)  # Zoll2 --> cm2
L2(12.9,cm2)
>>> L2(10,m2).to(yd2)    # m2 --> yd2
L2(11.96,yd2)
>>> meter2(3.5).to(yd2)  # m2 --> yd2 # globale Methode
L2(4.19,yd2)
>>> l24.to(m)            # m2 --> m ?
Traceback (most recent call last):
```



```
File "<stdin>", line 1, in <module>
File "D:\Python\units.py", line 5416, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: m
```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

4 Basisklasse L3

L3 realisiert das Rechnen mit Volumenmaßen.

4.1 Übersicht

Globale Informationen über die Klasse L3

Eine Übersicht über die Klasse L3 erhalten Sie durch die globale Methode `dok("L3")` oder – etwas ausführlicher – mit der L3-Methode `L3.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("L3.classInfo")`:

```
>>> dok("L3.classInfo")
L3.classInfo
    Gibt Informationen zur Klasse L3 und ihren Methoden aus.

    Aufruf: L3.classInfo(m=art) oder L3.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
    + "E"/Einheiten  : Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> L3.classInfo("H")
Class L3

    Realisiert das Rechnen mit L3-Instanzen (Volumenmaße).

    Aufruf: L3.__init__(value=1, unit=m3, n="")

    Parameter:

    value=1: Wert;
              Voreinstellung: 1
    unit=m3: L3-Maßeinheit;
              Voreinstellung: m3
    n=""    : Name der L3-Instanz;
              Voreinstellung: leere Zeichenkette
```

Methoden:

<code>a</code> :	L3-Instanz	
<code>n</code> :	Zahl(Integer/Float)	
<code>o</code> :	Instanz	
<code>i</code> :	Integer	
<code>u</code> :	eine der zulässigen L3-Maßeinheiten	
<code>a.__abs__()</code> oder <code>abs(a)</code>		Liefert den Absolut-Betrag der L3-Instanz.
<code>a.__add__(a)</code> oder <code>a + a</code>		Realisiert die Addition zweier L3-Instanzen.
<code>a.__eq__(a)</code> oder <code>a == a</code>		Realisiert den Vergleich <code>==</code> in L3.
<code>a.__ge__(a)</code> oder <code>a >= a</code>		Realisiert den Vergleich <code>>=</code> in L3.
<code>a.__gt__(a)</code> oder <code>a > a</code>		Realisiert den Vergleich <code>></code> in L3.
<code>a.__le__(a)</code> oder <code>a <= a</code>		Realisiert den Vergleich <code><=</code> in L3.
<code>a.__lt__(a)</code> oder <code>a < a</code>		Realisiert den Vergleich <code><</code> in L3.
<code>a.__mul__(o)</code> oder <code>a * o</code>		Realisiert die Multiplikation in L3.
<code>a.__neg__()</code> oder <code>-(a)</code>		Realisiert negatives Vorzeichen für eine L3-Instanz.
<code>a.__pos__()</code> oder <code>+(a)</code>		Realisiert positives Vorzeichen für eine L3-Instanz.
<code>a.__pow__(p)</code> oder <code>f ** p</code>		Realisiert das Potenzieren einer L3-Instanz.
<code>a.__repr__()</code> oder <code>repr(f)</code>		Repräsentiert eine L3-Instanz.
<code>a.__str__()</code> oder <code>str(f)</code>		Repräsentiert eine L3-Instanz.
<code>a.__sub__(a)</code> oder <code>f - a</code>		Realisiert die Subtraktion zweier L3-Instanzen.
<code>a.__truediv__(o)</code> oder <code>f / o</code>		Realisiert die Division einer L3-Instanz.
<code>L3.beispiel(e)</code>		Liefert Beispiele für die Methoden der Klasse L3.
<code>a.info(modus)</code>		Gibt Informationen über eine L3-Instanz aus.
<code>a.to(u)</code>		Realisiert die Konvertierung einer L3-Instanz in eine andere Maßeinheit.
<code>L3.__init__(value=1,unit=m3,n="")</code>		Initialisiert eine L3-Instanz.
<code>L3.classInfo(m=art)</code>		Gibt Informationen zur Klasse L3 und ihrer Methoden aus.
<code>L3.ci(m=art)</code>		Alias für <code>L3.classInfo(m=art)</code>
<code>L3.description()</code>		Gibt eine Kurzbeschreibung der Klasse L3 aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt 4.2 auf der nächsten Seite
- Maßeinheiten und Instanzen → Abschnitt 4.3 auf Seite 39
- Eigenschaften der Instanzen → Abschnitt 4.5 auf Seite 43
- Zulässige Operationen → Abschnitt 4.4 auf Seite 41
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 4.6 auf Seite 44

4.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("L3")` auf Seite 36 liefert u. a. eine Auflistung der L3-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 4.1 aufgeführt.

Tabelle 4.1: Methoden der Klasse L3

Name	Bedeutung
<code>L3.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse L3.
<code>L3.classInfo(art)</code>	Gibt Informationen zur Klasse L3 und ihrer Methoden aus.
<code>L3.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>L3.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse L3 aus.
<code>l.info(modus)</code>	Gibt Informationen über die L3-Instanz <i>l</i> aus.
<code>l.to(u)</code>	Realisiert die Konvertierung der L3-Instanz <i>l</i> in eine andere Maßeinheit <i>u</i> .

Ausführlichere Informationen über eine bestimmte L3-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `L3.classInfo("M")` für alle L3-Methoden.

Beispiele für dok

```
>>> dok("L3.to")
L3.to
    Realisiert die Konvertierung einer L3-Instanz in eine andere Maßeinheit.

    Aufruf: a.to(u)
    erlaubter Typ des Operanden:
    - eine der zulässigen L3-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("L3.info")
L3.info
    Gibt Informationen über eine L3-Instanz aus.

    Aufruf: a.info(m)
    m Modus
    kurz   nur Grund-Eigenschaften
           [Voreinstellung]
    lang   auch Darstellung in anderen Einheiten
```

4.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die Maßeinheiten in L3 liefert die globalen Methoden `allUnits("L3")` und `au("L3")`:

```
>>> allUnits("L3")
Volumenmaße (in m3): 21 Einträge
  bbl      0.16      amerik. Volumen-Maß: blue barrel (Erdöl);
                        42 US.liq.gal
  ccm      0.0       Volumen-Maß: (cm) hoch 3
  cl       0.0       Volumen-Maß: Zentiliter (cl): 1e-2 Liter
  cm3      0.0       Volumen-Maß: (cm) hoch 3
  dl       0.0       Volumen-Maß: Deziliter (dl): 1e-1 Liter
  dm3      0.0       Volumen-Maß: (dm) hoch 3
  fl       0.0       Volumen-Maß: Femtoliter (fl): 1e-15 Liter
  ft3      0.03      amerik. Volumen-Maß: cubic foot; 1728 cubic inch
  gal      0.0       amerik. Volumen-Maß: US.liq.gal; 231 cubic inch
  hl       0.1       Volumen-Maß: Hektoliter (hl): 100 liter
  inch3    0.0       amerik. Volumen-Maß: cubic inch
  km3      1000000000.0 Volumen-Maß: (km) hoch 3
  liter    0.0       Volumen-Maß: 1 dm3
  m3       1         Volumen-Maß: m hoch 3
  ml       0.0       Volumen-Maß: Milliliter (ml): 1e-3 Liter
  mm3      0.0       Volumen-Maß: (mm) hoch 3
  myl      0.0       Volumen-Maß: Mikroliter (myl): 1e-6 Liter
  nl       0.0       Volumen-Maß: Nanoliter (nl): 1e-9 Liter
  pint     0.0       amerik. Volumen-Maß: US.liq.pt; US.liq.gal / 8
  pl       0.0       Volumen-Maß: Pikoliter (pl): 1e-12 Liter
  yd3      0.76      amerik. Volumen-Maß: cubic yard
```

Interessant sind die amerikanischen Volumenmaße (`ft3`, `gal`, `inch3`, `pint`, `yd3`).

Instanzen

Eigene L3-Instanzen können Sie mittels der Anweisung `L3(volumen, einheit, n=name)` oder der globalen Methode `meter3(wert)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> l31 = L3(2, m3)
>>> l32 = L3(-2, mm3)
>>> l33 = -L3(0.5, km3)
>>> l34 = +meter3(3.750) # globale Methode
>>> l31
L3(2,m3)
>>> l32
```

```

L3(-2,mm3)
>>> 133
L3(-0.5,km3)
>>> 134
L3(3.75,m3)

```

Mögliche Fehlerquellen

Der Versuch, eine eigene L3-Instanz zu generieren, kann misslingen:

- **Unit ist hier nicht bekannt/zulässig:** Nur die bei einem Aufruf der globalen Methode `allUnits("L3")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("L3.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne L3-Instanz erhalten Sie mit der L3-Methode `info`, beispielsweise für 131:

```

>>> 131.info()
Name          : L3(2,m3)
Art           : L3-Instanz (Volumenmaß)
Wert          : 2 (in m3)
Einheit       : m3 [Volumen-Maß: m hoch 3]
interner Wert: 2 (in m3)
-----

```

Informationen über alle L3-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```

>>> alle("L3")
7 Element(e):

Name          : L1(-0.5,km) ** (3)
Art           : L3-Instanz (Volumenmaß)
Wert          : -125000000.0 (in m3)
Einheit       : m3 [Volumen-Maß: m hoch 3]
interner Wert: -125000000.0 (in m3)
-----

Name          : L3(2,m3)
Art           : L3-Instanz (Volumenmaß)
Wert          : 2 (in m3)
Einheit       : m3 [Volumen-Maß: m hoch 3]
interner Wert: 2 (in m3)
-----

Name          : L3(-2,mm3)
Art           : L3-Instanz (Volumenmaß)

```

```
Wert          : -2 (in mm3)
Einheit       : mm3 [Volumen-Maß: (mm) hoch 3]
interner Wert: -0.0 (in m3)
-----
```

```
Name         : L3(0.5,km3)
Art          : L3-Instanz (Volumenmaß)
Wert         : 0.5 (in km3)
Einheit      : km3 [Volumen-Maß: (km) hoch 3]
interner Wert: 500000000.0 (in m3)
-----
```

```
Name         : L3(-0.5,km3)
Art          : L3-Instanz (Volumenmaß)
Wert         : -0.5 (in km3)
Einheit      : km3 [Volumen-Maß: (km) hoch 3]
interner Wert: -500000000.0 (in m3)
-----
```

```
Name         : meter3(3.75)
Art          : L3-Instanz (Volumenmaß)
Wert         : 3.75 (in m3)
Einheit      : m3 [Volumen-Maß: m hoch 3]
interner Wert: 3.75 (in m3)
-----
```

```
Name         : L3(3.75,m3)
Art          : L3-Instanz (Volumenmaß)
Wert         : 3.75 (in m3)
Einheit      : m3 [Volumen-Maß: m hoch 3]
interner Wert: 3.75 (in m3)
-----
```

4.4 Operatoren

Vergleiche

Instanzen der Klasse L3 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> 131, 132, 133, 134
(L3(2,m3), L3(-2,mm3), L3(-0.5,km3), L3(3.75,m3))
>>> 131 <= 132
False
>>> 133 >= 134
False
>>> 132 != L3(23, m3)
True
>>> 134 == L3(3.4, cm3)
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> 131
L3(2,m3)
>>> (L3(1, m3) <= 131) and (131 <= L3(5, m3))      # 131 zwischen 1m3 und 5m3
True
>>> not ((131 == L3(1, m3)) or (131 == L3(5, m3))) # 131 weder 1m3 noch 5m3
True

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse L3 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 4.2

Tabelle 4.2: Klasse L3, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	L3	G	G	G	G	G	G	L3
/	G	G	G	L2	L1	Skalar	G	L3	G	G	G	G	G	G	L3
+	–	–	–	–	–	–	–	L3	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	L3	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```

>>> l31, l32, l33, l34
(L3(2,m3), L3(-2,mm3), L3(-0.5,km3), L3(3.75,m3))
>>> l31 + l32
L3(2.0,m3)
>>> l33 - L3(23, m3)
L3(-500000023.0,m3)
>>> l34 * L3(3.4, cm3)
G(L3(3.75,m3) * (L3(3.4,cm3)))
>>> l31 / l32
-999999999.9999999
>>> l33 ** (2/3)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'L3' and 'float'

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit L3-Instanzen erhalten Sie durch `L3.beispiel(beispiel)`. Welche Angaben für *beispiel* möglich sind, erfahren Sie durch die globale Methode `dok("L3.beispiel")`, beispielsweise `L3.beispiel()`.

4.5 Eigenschaften der L3-Instanzen

Jede L3-Instanz (jedes L3-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der L3-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `L3.classInfo("V")`. Danach stehen in der Klasse L3 die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> L3.classInfo("V")
Eigenschaften der L3-Instanzen:

l.v          Wert der Instanz (gemessen in l.u)
l.u          Maßeinheit der Instanz (L3-Maßeinheit)
l.name       Name der Instanz
l.internal   interner Wert

```

Beispiele

```

>>> l34
L3(3.75,m3)
>>> l34.info()

```

```

Name          : L3(3.75,m3)
Art           : L3-Instanz (Volumenmaß)
Wert          : 3.75 (in m3)
Einheit       : m3 [Volumen-Maß: m hoch 3]
interner Wert: 3.75 (in m3)
-----

```

```

>>> l34.v
3.75
>>> l34.u
'm3'
>>> l34.name
'L3(3.75,m3)'
>>> l34.internal
3.75

```

4.6 Konvertierung

Instanzen der Klasse L3 können mit der Hilfe der L3-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden:

Beispiele – Konvertierung

```

>>> l31, l32, l33, l34
(L3(2,m3), L3(-2,mm3), L3(-0.5,km3), L3(3.75,m3))
>>> l31.to(km3)          # m3 --> km3
L3(0.0,km3)
>>> l32.to(m3)           # mm3 --> m3
L3(-0.0,m3)
>>> l33.to(cm3)          # km3 --> cm3
L3(-5000000000000000.0,cm3)
>>> l34.to(mm3)          # m3 --> mm3
L3(3750000000.0,mm3)
>>> L3(2,inch3).to(cm3)  # Zoll3 --> cm3
L3(32.77,cm3)
>>> L3(10,m3).to(yd3)    # m3 --> yd3
L3(13.08,yd3)
>>> meter3(3.5).to(yd3)  # m3 --> yd3 # globale Methode
L3(4.58,yd3)
>>> l34.to(m2)           # m3 --> m2 ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 5949, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: m2

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

5 Basisklasse T1

T1 realisiert das Rechnen mit Zeitmaßen.

5.1 Übersicht

Globale Informationen über die Klasse T1

Eine Übersicht über die Klasse **T1** erhalten Sie durch die globale Methode `dok("T1")` oder – etwas ausführlicher – mit der **T1**-Methode `T1.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("T1.classInfo")`:

```
>>> dok("T1.classInfo")
T1.classInfo
    Gibt Informationen zur Klasse T1 und ihre Methoden aus.

    Aufruf: T1.classInfo(m=art) oder T1.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen: Variablen/Eigenschaften
    + "M"/Methoden  : Methoden
    + "E"/Einheiten : Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> T1.classInfo("H")
Class T1

    Realisiert das Rechnen mit T1-Instanzen (Zeiten).

    Aufruf: T1.__init__(value=1, unit=s, n="")

    Parameter:

    value=1 : Wert;
              Voreinstellung: 1
    unit=s  : T1-Maßeinheit;
              Voreinstellung: s
    n=""    : Name der T1-Instanz;
              Voreinstellung: leere Zeichenkette
```

Methoden:

```

t : T1-Instanz (Zeit)
o : Instanz
n : Zahl(Integer/Float)
p : Integer
u : eine der zulässigen T1-Maßeinheiten

t.__abs__() oder abs(t)      Liefert den Absolut-Betrag der T1-Instanz.
t.__add__(t) oder t + t     Realisiert die Addition zweier T1-Instanzen.
t.__eq__(t) oder t == t     Realisiert den Vergleich == in T1.
t.__ge__(t) oder t >= t     Realisiert den Vergleich >= in T1.
t.__gt__(t) oder t > t      Realisiert den Vergleich > in T1.
t.__le__(t) oder t <= t     Realisiert den Vergleich <= in T1.
t.__lt__(t) oder t < t      Realisiert den Vergleich < in T1.
t.__mul__(o) oder t * o     Realisiert die Multiplikation.
t.__ne__(t) oder t != t     Realisiert den Vergleich != in T1.
t.__neg__() oder -(t)       Realisiert negatives Vorzeichen in T1.
t.__pos__() oder +(t)       Realisiert positives Vorzeichen in T1.
t.__pow__(p) oder t ** p    Realisiert das Potenzieren einer T1-Instanz.
t.__repr__() oder repr(t)   Repräsentiert eine T1-Instanz.
t.__str__() oder str(t)     Repräsentiert eine T1-Instanz.
t.__sub__(t) oder t - t     Realisiert die Subtraktion in T1.
t.__truediv__(o) oder t / o Realisiert die Division einer T1-Instanz.
t.info(modus)               Gibt Informationen über eine T1-Instanz aus.
t.to(u)                     Realisiert die Konvertierung einer T1-Instanz in
                             eine andere Maßeinheit.
t.toYMD()                   Liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde
                             der aktuellen T1-Instanz als Liste.
t.toU()                     Konvertiert die aktuelle Zeit in die Uhrzeit U.
T1.__init__(value=1,unit=s,n="") Initialisiert eine T1-Instanz.
T1.beispiel(e)              Liefert Beispiele für die Methoden der Klasse T1.
T1.classInfo(m=art)         Gibt Informationen zur Klasse T1 und ihre
                             Methoden aus.
T1.ci(m=art)                Alias für T1.classInfo(m=art)
T1.description()            Gibt eine Kurzbeschreibung der Klasse T1 aus.
-----

```

Weitere Informationen

- Methoden der Klasse → Abschnitt 5.2 auf der nächsten Seite
- Maßeinheiten und Instanzen → Abschnitt 5.3 auf Seite 48
- Eigenschaften der Instanzen → Abschnitt 5.5 auf Seite 52
- Zulässige Operationen → Abschnitt 5.4 auf Seite 50
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 5.6 auf Seite 53

5.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("T1")` auf Seite 45 liefert u. a. eine Auflistung der T1-Methoden. Namen, die die Zeichenfolge `--` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 5.1 aufgeführt.

Tabelle 5.1: Methoden der Klasse T1

Name	Bedeutung
<code>T1.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse T1.
<code>T1.classInfo(art)</code>	Gibt Informationen zur Klasse T1 und ihre Methoden aus.
<code>T1.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>T1.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse T1 aus.
<code>t.info(modus)</code>	Gibt Informationen über die T1-Instanz <i>t</i> aus.
<code>t.toU()</code>	Konvertiert die aktuelle Zeit <i>t</i> in die Uhrzeit U.
<code>t.to(u)</code>	Realisiert die Konvertierung der T1-Instanz <i>t</i> in eine andere Maßeinheit <i>u</i> .
<code>t.toYMD()</code>	Liefert <i>Jahr</i> , <i>Monat</i> , <i>Tag</i> , <i>Stunde</i> , <i>Minute</i> , <i>Sekunde</i> der aktuellen T1-Instanz <i>t</i> als Liste.

Ausführlichere Informationen über eine bestimmte T1-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `T1.classInfo("M")` für alle T1-Methoden.

Beispiele für dok

```
>>> dok("T1.info")
T1.info
    Gibt Informationen über eine T1-Instanz aus.

    Aufruf: t.info(modus)
    m Modus
        kurz  nur Grund-Eigenschaften
              [Voreinstellung]
        lang  auch Darstellung in anderen Einheiten
>>> dok("T1.to")
T1.to
    Realisiert die Konvertierung einer T1-Instanz in eine andere Maßeinheit.

    Aufruf: t.to(unit)
    erlaubter Typ des Operanden:
    - eine der zulässigen T1-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("T1.toYMD")
T1.toYMD
```

Liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde der aktuellen Instanz als Liste.

Aufruf: `t.toYMD()`

```
>>> dok("T1.toU")
```

T1.toU

Konvertiert die aktuelle Zeit in die Uhrzeit U.

Aufruf: `t.toU()`

mögliche Fehlermeldung:

- Parameter ist unzulässig

5.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die Maßeinheiten in T1 liefern die globalen Methoden `allUnits("T1")` und `au("T1")`:

```
>>> allUnits("T1")
```

Zeitmaße (in s): 12 Einträge

d	86400	Zeitmaß: mittlerer Sonnentag; 86400 s
fs	0.0	Zeitmaß: Femtosekunde; 1e-15 s
h	3600	Zeitmaß: Stunde; 3600 s
minute	60	Zeitmaß: Minute; 60 s
mon	2592000	Zeitmaß: gesetzlicher Monat (month), 30 d; 2592000 s
ms	0.0	Zeitmaß: Millisekunde; 1e-3 s
mys	0.0	Zeitmaß: Mikrosekunde; 1e-6 s
ns	0.0	Zeitmaß: Nanosekunde; 1e-9 s
ps	0.0	Zeitmaß: Picosekunde; 1e-12 s
s	1	Zeitmaß: Sekunde; 1 s
week	604800	Zeitmaß: Woche, 7 d; 604800 s
y	31536000	Zeitmaß: Kalenderjahr (year), 365 d; 31536000 s

Instanzen

Eigene T1-Instanzen können Sie mittels der Anweisung `T1(zeit, einheit, n=name)` oder der globalen Methode `sekunde(wert)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> t11 = T1(2, s)
>>> t12 = T1(-2, d)
>>> t13 = -T1(0.5, h)
>>> l0 = [0, 0, 13, 4, 5.7] # [zy, zmo, zd, zh, zmi, zs]
>>> t5 = ymdToT1(l0) # globale Methode; [zy, zmo, zd, zh, zmi, zs] ---> T1
>>> t14 = sekunde(3.750) # globale Methode
```

```
>>> t11, t12, t13, t14, t5
(T1(2,s), T1(-2,d), T1(-0.5,h), T1(3.75,s), T1(1137942.0,s))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene T1-Instanz zu generieren, kann misslingen:

- `Unit` ist hier nicht bekannt/zulässig: Nur die bei einem Aufruf der globalen Methode `allUnits("T1")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("T1.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne T1-Instanz erhalten Sie mit der T1-Methode `info`, beispielsweise für `t12`:

```
>>> t12.info()
Name          : T1(-2,d)
Art           : T1-Instanz (Zeit)
Wert          : -2 (in d)
Einheit       : d [Zeitmaß: mittlerer Sonnentag; 86400 s]
interner Wert: -172800 (in s)
-----
```

Informationen über alle T1-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("T1")
6 Element(e):

Name          : T1(2,s)
Art           : T1-Instanz (Zeit)
Wert          : 2 (in s)
Einheit       : s [Zeitmaß: Sekunde; 1 s]
interner Wert: 2 (in s)
-----

Name          : T1(-2,d)
Art           : T1-Instanz (Zeit)
Wert          : -2 (in d)
Einheit       : d [Zeitmaß: mittlerer Sonnentag; 86400 s]
interner Wert: -172800 (in s)
-----

Name          : T1(0.5,h)
Art           : T1-Instanz (Zeit)
Wert          : 0.5 (in h)
Einheit       : h [Zeitmaß: Stunde; 3600 s]
interner Wert: 1800.0 (in s)
```

```

-----
Name      : T1(-0.5,h)
Art       : T1-Instanz (Zeit)
Wert      : -0.5 (in h)
Einheit   : h [Zeitmaß: Stunde; 3600 s]
interner Wert: -1800.0 (in s)
-----

```

```

-----
Name      : T1(1137942.0,s)
Art       : T1-Instanz (Zeit)
Wert      : 1137942.0 (in s)
Einheit   : s [Zeitmaß: Sekunde; 1 s]
interner Wert: 1137942.0 (in s)
-----

```

```

-----
Name      : sekunde(3.75)
Art       : T1-Instanz (Zeit)
Wert      : 3.75 (in s)
Einheit   : s [Zeitmaß: Sekunde; 1 s]
interner Wert: 3.75 (in s)
-----

```

5.4 Operatoren

Vergleiche

Instanzen der Klasse T1 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> t11, t12, t13, t14
(T1(2,s), T1(-2,d), T1(-0.5,h), T1(3.75,s))
>>> t11 <= t12
False
>>> t13 >= t14
False
>>> t12 != T1(23, s)
True

```



```
>>> t14 == T1(3.4, h)
False
```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> t11
T1(2,s)
>>> (T1(1, s) <= t11) and (t11 <= T1(5, s))      # t11 zwischen 1s und 5s
True
>>> not ((t11 == T1(1, s)) or (t11 == T1(5, s))) # t11 weder 1s noch 5s
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse T1 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 5.2.

Tabelle 5.2: Klasse T1, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	V	G	G	G	G	G	T1	G	W	T2	G	G	G	G	T1
/	G	G	G	G	G	G	G	T1	G	Skalar	G	G	G	G	T1
+	–	–	–	–	–	–	–	T1	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	T1	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```
>>> t11, t12, t13, t14
(T1(2,s), T1(-2,d), T1(-0.5,h), T1(3.75,s))
>>> t11 + t12
T1(-172798,s)
>>> t13 - T1(23, s)
T1(-1823.0,s)
>>> t14 * T(3.4, h)
T2(45900.0,s2)
>>> t11 / t12
-1.1574074074074073e-05
>>> t13 ** 2
T2(3240000.0,s2)
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit T1-Instanzen erhalten Sie durch den Aufruf der T1-Methode `T1.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("T1.beispiel")`, beispielsweise `T1.beispiel()`.

5.5 Eigenschaften der T1-Instanzen

Jede T1-Instanz (jedes T1-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der T1-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `T1.classInfo("V")`. Danach stehen in der Klasse T1 die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> T1.classInfo("V")
```

Eigenschaften der T1-Instanzen:

```
t.v          Wert der Instanz (gemessen in t.u)
t.u          Maßeinheit der Instanz (T1-Maßeinheit)
t.name       Name der Instanz
t.internal   interner Wert
```

Beispiele

```
>>> t13
```

```
T1(-0.5,h)
```

```
>>> t13.info()
```

```
Name          : T1(-0.5,h)
Art           : T1-Instanz (Zeit)
Wert          : -0.5 (in h)
Einheit       : h [Zeitmaß: Stunde; 3600 s]
interner Wert: -1800.0 (in s)
```

```
-----
```

```
>>> t13.v
```

```
-0.5
```

```
>>> t13.u
```

```
'h'
```

```
>>> t13.name
```

```
'T1(-0.5,h)'
```

```
>>> t13.internal
```

```
-1800.0
```

5.6 Konvertierung

Instanzen der Klasse T1 können mit der Hilfe der T1-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden. Ein Aufruf der T1-Methode `toYMD` liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde der aktuellen T1-Instanz als Liste.

Beispiele – Konvertierung

```
>>> t11, t12, t13, t14
(T1(2,s), T1(-2,d), T1(-0.5,h), T1(3.75,s))
>>> t11.to(h)           # s --> h
T1(0.0,h)
>>> t12.to(h)           # d --> h
T1(-48.0,h)
>>> t13.to(minute)      # h --> minute
T1(-30.0,minute)
>>> t14.to(h)           # s --> h
T1(0.0,h)
>>> T1(2,week).to(h)    # week --> h
T1(336.0,h)
>>> T1(10,minute).to(s) # minute --> s
T1(600.0,s)
>>> t12.toYMD           # Zeit --> YMD (year, month, day, ...)
<bound method T1.toYMD of T1(-2,d)>
>>> sekunde(350).to(h)  # s --> h    # globale Methode
T1(0.1,h)
>>> t14.to(s2)          # s --> s2 ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 7538, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: s2
```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

6 Basisklasse T2

T2 realisiert das Rechnen mit Zeiteinheiten².

6.1 Übersicht

Globale Informationen über die Klasse T2

Eine Übersicht über die Klasse T2 erhalten Sie durch die globale Methode `dok("T2")` oder – etwas ausführlicher – mit der T2-Methode `T2.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("T2.classInfo")`:

```
>>> dok("T2.classInfo")
T2.classInfo
    Gibt Informationen zur Klasse T2 und ihre Methoden aus.

    Aufruf: T2.classInfo(m=art) oder T2.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen: Variablen/Eigenschaften
    + "M"/Methoden  : Methoden
    + "E"/Einheiten: Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> T2.classInfo("H")
Class T2

    Realisiert das Rechnen mit T2-Instanzen (Zeiten hoch 2).

    Aufruf: T2.__init__(value=1, unit=s2, n="")

    Parameter:

    value=1: Wert;
              Voreinstellung: 1
    unit=s2: T2-Maßeinheit;
              Voreinstellung: s2
    n=""    : Name der T2-Instanz;
              Voreinstellung: leere Zeichenkette
```

Methoden:

```

t : T2-Instanz (Zeit hoch 2)
o : Instanz
n : Zahl(Integer/Float)
p : Integer
u : eine der zulässigen T2-Maßeinheiten

t.__abs__() oder abs(t)      Liefert den Absolut-Betrag der T2-Instanz.
t.__add__(t) oder t + t     Realisiert die Addition zweier T2-Instanzen.
t.__eq__(t) oder t == t     Realisiert den Vergleich == in T2.
t.__ge__(t) oder t >= t     Realisiert den Vergleich >= in T2.
t.__gt__(t) oder t > t      Realisiert den Vergleich > in T2.
t.__le__(t) oder t <= t     Realisiert den Vergleich <= in T2.
t.__lt__(t) oder t < t      Realisiert den Vergleich < in T2.
t.__mul__(o) oder t * o     Realisiert die Multiplikation einer T2-Instanz.
t.__ne__(t) oder t != t     Realisiert den Vergleich != in T2.
t.__neg__() oder -(t)       Realisiert negatives Vorzeichen für eine
                             T2-Instanz.
t.__pos__() oder +(t)       Realisiert positives Vorzeichen für eine
                             T2-Instanz.
t.__pow__(p) oder t ** p    Realisiert das Potenzieren einer T2-Instanz.
t.__repr__() oder repr(t)   Repräsentiert eine T2-Instanz.
t.__str__() oder str(t)     Repräsentiert eine T2-Instanz.
t.__sub__(t) oder t - t     Realisiert die Subtraktion.
t.__truediv__(o) oder t / o Realisiert die Division einer T2-Instanz.
t.info(modus)               Gibt Informationen über eine T2-Instanz aus.
t.to(u)                     Realisiert die Konvertierung einer T2-Instanz in
                             eine andere Maßeinheit.
T2.__init__(value=1,unit=s2,n="") Initialisiert eine T2-Instanz.
T2.beispiel(e)               Liefert Beispiele für die Methoden der Klasse T2.
T2.classInfo(m=art)         Gibt Informationen zur Klasse T2 und ihre Methoden
                             aus.
T2.ci(m=art)                 Alias für T2.classInfo(m=art)
T2.description()             Gibt eine Kurzbeschreibung der Klasse T2 aus.

```

Weitere Informationen

- Methoden der Klasse → Abschnitt 6.2 auf der nächsten Seite
- Maßeinheiten und Instanzen → Abschnitt 6.3 auf Seite 57
- Eigenschaften der Instanzen → Abschnitt 6.5 auf Seite 61
- Zulässige Operationen → Abschnitt 6.4 auf Seite 59
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 6.6 auf Seite 62

6.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("T2")` auf Seite 54 liefert u. a. eine Auflistung der T2-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 6.1 aufgeführt.

Tabelle 6.1: Methoden der Klasse T2

Name	Bedeutung
<code>T2.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse T2.
<code>T2.classInfo(art)</code>	Gibt Informationen zur Klasse T2 und ihre Methoden aus.
<code>T2.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>T2.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse T2 aus.
<code>t.info(modus)</code>	Gibt Informationen über die T2-Instanz <i>t</i> aus.
<code>t.to(u)</code>	Realisiert die Konvertierung der T2-Instanz <i>t</i> in eine andere Maßeinheit <i>u</i> .

Ausführlichere Informationen über eine bestimmte T2-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `T2.classInfo("M")` für alle T2-Methoden.

Beispiele für dok

```
>>> dok("T2.to")
T2.to
    Realisiert die Konvertierung einer T2-Instanz in eine andere Maßeinheit.

    Aufruf: t.to(unit)
    erlaubter Typ des Operanden:
    - eine der zulässigen T2-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("T2.info")
T2.info
    Gibt Informationen über eine T2-Instanz aus.

    Aufruf: t.info(modus)
    m Modus
    kurz   nur Grund-Eigenschaften
           [Voreinstellung]
    lang   auch Darstellung in anderen Einheiten
```

6.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die Maßeinheiten in T2 liefern die globalen Methoden `allUnits("T2")` und `au("T2")`:

```
>>> allUnits("T2")
Zeitmaße hoch 2 (in s2): 12 Einträge
d2      7464960000      Zeitmaß hoch 2: (Sonnentag) hoch 2;
                        (86400 s) hoch 2
fs2      0.0           Zeitmaß hoch 2: (Femtosekunde) hoch 2;
                        (1e-15 s) hoch 2
h2       12960000       Zeitmaß hoch 2: (Stunde) hoch 2; (3600 s) hoch 2
minute2  3600          Zeitmaß hoch 2: (Minute) hoch 2; (60 s) hoch 2
mon2     6718464000000  Zeitmaß hoch 2: (Monat) hoch 2;
                        (2592000 s) hoch 2
ms2      0.0           Zeitmaß hoch 2: (Millisekunde) hoch 2;
                        (1e-3 s) hoch 2
mys2     0.0           Zeitmaß hoch 2: (Mikrosekunde) hoch 2;
                        (1e-6 s) hoch 2
ns2      0.0           Zeitmaß hoch 2: (Nanosekunde) hoch 2;
                        (1e-9 s) hoch 2
ps2      0.0           Zeitmaß hoch 2: (Picosekunde) hoch 2;
                        (1e-12 s) hoch 2
s2       1             Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2
week2    365783040000   Zeitmaß hoch 2: (Woche) hoch 2; (604800 s) hoch 2
y2       994519296000000 Zeitmaß hoch 2: (Kalenderjahr) hoch 2;
                        (31536000 s) hoch 2
```

Instanzen

Eigene T2-Instanzen können Sie mittels der Anweisung `T2(zeit2, einheit, n=name)` oder der globalen Methode `sekunde2(wert)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> t21 = T2(2, s2)
>>> t22 = T2(-2, d2)
>>> t23 = -T2(0.5, h2)
>>> t24 = +sekunde2(3.750) # globale Methode
>>> t21, t22, t23, t24
(T2(2,s2), T2(-2,d2), T2(-0.5,h2), T2(3.75,s2))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene T2-Instanz zu generieren, kann misslingen:

- **Unit ist hier nicht bekannt/zulässig:** Nur die bei einem Aufruf der globalen Methode `allUnits("T2")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("T2.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne T2-Instanz erhalten Sie mit der T2-Methode `info`, beispielsweise für `t23`:

```
>>> t23.info()
Name          : T2(-0.5,h2)
Art           : T2-Instanz (Zeit hoch 2)
Wert          : -0.5 (in h2)
Einheit       : h2 [Zeitmaß hoch 2: (Stunde) hoch 2; (3600 s) hoch 2]
interner Wert: -6480000.0 (in s2)
-----
```

Informationen über alle T2-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("T2")
8 Element(e):

Name          : T1(3.75,s) * (T1(3.4,h))
Art           : T2-Instanz (Zeit hoch 2)
Wert          : 45900.0 (in s2)
Einheit       : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 45900.0 (in s2)
-----

Name          : T1(-0.5,h) ** (2)
Art           : T2-Instanz (Zeit hoch 2)
Wert          : 3240000.0 (in s2)
Einheit       : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 3240000.0 (in s2)
-----

Name          : T2(2,s2)
Art           : T2-Instanz (Zeit hoch 2)
Wert          : 2 (in s2)
Einheit       : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 2 (in s2)
-----

Name          : T2(-2,d2)
Art           : T2-Instanz (Zeit hoch 2)
Wert          : -2 (in d2)
Einheit       : d2 [Zeitmaß hoch 2: (Sonntag) hoch 2;
```



```

                                (86400 s) hoch 2]
interner Wert: -14929920000 (in s2)
-----

Name      : T2(0.5,h2)
Art       : T2-Instanz (Zeit hoch 2)
Wert      : 0.5 (in h2)
Einheit   : h2 [Zeitmaß hoch 2: (Stunde) hoch 2; (3600 s) hoch 2]
interner Wert: 6480000.0 (in s2)
-----

Name      : T2(-0.5,h2)
Art       : T2-Instanz (Zeit hoch 2)
Wert      : -0.5 (in h2)
Einheit   : h2 [Zeitmaß hoch 2: (Stunde) hoch 2; (3600 s) hoch 2]
interner Wert: -6480000.0 (in s2)
-----

Name      : sekunde2(3.75)
Art       : T2-Instanz (Zeit hoch 2)
Wert      : 3.75 (in s2)
Einheit   : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 3.75 (in s2)
-----

Name      : T2(3.75,s2)
Art       : T2-Instanz (Zeit hoch 2)
Wert      : 3.75 (in s2)
Einheit   : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 3.75 (in s2)
-----

```

6.4 Operatoren

Vergleiche

Instanzen der Klasse T2 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> t21, t22, t23, t24
(T2(2,s2), T2(-2,d2), T2(-0.5,h2), T2(3.75,s2))
>>> t21 <= t22
False
>>> t23 >= t24
False
>>> t22 != T2(23, s2)
True
>>> t24 == T2(3.4, h2)
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> t21
T2(2,s2)
>>> (T2(1, s2) <= t21) and (t21 <= T2(5, s2))      # t21 zwischen 1s2 und 5s2
True
>>> not ((t21 == T2(1, s2)) or (t21 == T2(5, s2))) # t21 weder 1s2 noch 5s2
True

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse T2 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 6.2.

Tabelle 6.2: Klasse T2, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	T2	G	G	G	G	G	G	T2
/	G	G	G	G	G	G	G	T2	G	T1	Skalar	G	G	G	T2
+	–	–	–	–	–	–	–	T2	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	T2	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```

>>> t21, t22, t23, t24
(T2(2,s2), T2(-2,d2), T2(-0.5,h2), T2(3.75,s2))
>>> t21 + t22
T2(-14929919998,s2)
>>> t23 - T2(23, s2)
T2(-6480023.0,s2)
>>> t24 * T2(3.4, h2)
G(T2(3.75,s2) * (T2(3.4,h2)))
>>> t21 / t22
-1.3395919067215363e-10
>>> t23 ** 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'T2' and 'int'

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit T2-Instanzen erhalten Sie durch den Aufruf der T2-Methode `T2.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("T2.beispiel")`, beispielsweise `T2.beispiel()`.

6.5 Eigenschaften der T2-Instanzen

Jede T2-Instanz (jedes T2-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der T2-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `T2.classInfo("V")`. Danach stehen in der Klasse T2 die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> T2.classInfo("V")
Eigenschaften der T2-Instanzen:

t.v          Wert der Instanz (gemessen in t.u)
t.u          Maßeinheit der Instanz (T2-Maßeinheit)
t.name       Name der Instanz
t.internal   interner Wert

```

Beispiele

```

>>> t24
T2(3.75,s2)
>>> t24.info()

```

```

Name          : T2(3.75,s2)
Art           : T2-Instanz (Zeit hoch 2)
Wert          : 3.75 (in s2)
Einheit       : s2 [Zeitmaß hoch 2: (Sekunde) hoch 2; (1 s) hoch 2]
interner Wert: 3.75 (in s2)
-----

```

```

>>> t24.v
3.75
>>> t24.u
's2'
>>> t24.name
'T2(3.75,s2)'
>>> t24.internal
3.75

```

6.6 Konvertierung

Instanzen der Klasse T2 können mit der Hilfe der T2-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden:

Beispiele – Konvertierung

```

>>> t21, t22, t23, t24
(T2(2,s2), T2(-2,d2), T2(-0.5,h2), T2(3.75,s2))
>>> t21.to(minute2)      # s2 --> minute2
T2(0.0,minute2)
>>> t22.to(minute2)      # d2 --> minute2
T2(-4147200.0,minute2)
>>> t23.to(s2)           # h2 --> s2
T2(-6480000.0,s2)
>>> t24.to(h2)           # s2 --> h2
T2(0.0,h2)
>>> sekunde2(350).to(h2) # s2 --> h2 # globale Methode
T2(0.0,h2)
>>> t24.to(s)            # s2 --> s ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 8524, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: s

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

7 Basisklasse M

M realisiert das Rechnen mit Massenmaßen/Gewichtsmaßen.

7.1 Übersicht

Globale Informationen über die Klasse M

Eine Übersicht über die Klasse M erhalten Sie durch die globale Methode `dok("M")` oder – etwas ausführlicher – mit der M-Methode `M.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("M.classInfo")`:

```
>>> dok("M.classInfo")
M.classInfo
    Gibt Informationen zur Klasse M und ihre Methoden aus.

    Aufruf: M.classInfo(m=art) oder M.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen: Variablen/Eigenschaften
    + "M"/Methoden  : Methoden
    + "E"/Einheiten; Einheiten
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> M.classInfo("H")
Class M

    Realisiert das Rechnen mit M-Instanzen (Gewichtsmaße).

    Aufruf: M.__init__(value=1, unit=kg, n="")

    Parameter:

    value=1 : Wert;
              Voreinstellung: 1
    unit=kg : M-Maßeinheit;
              Voreinstellung: kg
    n=""    : Name der M-Instanz;
              Voreinstellung: leere Zeichenkette
```

Methoden:

```

m : M-Instanz (Gewicht)
n : Zahl(Integer/Float)
p : Integer
o : Instanz
u : eine der zulässigen M-Maßeinheiten

m.__abs__() oder abs(m)      Liefert den Absolut-Betrag der M-Instanz.
m.__add__(m) oder m + m     Realisiert die Addition zweier M-Instanzen.
m.__eq__(m) oder m == m     Realisiert den Vergleich == in M.
m.__ge__(m) oder m >= m     Realisiert den Vergleich >= in M.
m.__gt__(m) oder m > m      Realisiert den Vergleich > in M.
M.__init__(value=1,unit=kg,n="") Initialisiert eine M-Instanz.
m.__le__(m) oder m <= m     Realisiert den Vergleich <= in M.
m.__lt__(m) oder m < m      Realisiert den Vergleich < in M.
m.__mul__(o) oder m * o     Realisiert die Multiplikation einer M-Instanz.
m.__ne__(m) oder m != m     Realisiert den Vergleich != in M.
m.__neg__() oder -(m)       Realisiert negatives Vorzeichen in M.
m.__pos__() oder +(m)       Realisiert positives Vorzeichen in M.
m.__pow__(p) oder m ** p    Realisiert das Potenzieren einer M-Instanz.
m.__repr__() oder repr(m)   Repräsentiert eine M-Instanz.
m.__str__() oder str(m)     Repräsentiert eine M-Instanz.
m.__sub__(m) oder m - m     Realisiert die Subtraktion zweier M-Instanzen.
m.__truediv__(o) oder m / o Realisiert die Division einer M-Instanz.
M.beispiel(e)               Liefert Beispiele für die Methoden der Klasse M.
M.classInfo(m=art)          Gibt Informationen zur Klasse M und ihre Methoden aus.

M.ci(m=art)                 Alias für M.classInfo(m=art)
M.description()             Gibt eine Kurzbeschreibung der Klasse M aus.
m.info(modus)               Gibt Informationen über eine M-Instanz aus.
m.to(u)                     Realisiert die Konvertierung einer M-Instanz in eine andere Maßeinheit.

m.toLongweights()           Liefert lton, cwt, qu, stone, lb, oz, dr, grain der aktuellen Instanz als Liste.

```

Weitere Informationen

- Methoden der Klasse → Abschnitt [7.2 auf der nächsten Seite](#)
- Maßeinheiten und Instanzen → Abschnitt [7.3 auf Seite 66](#)
- Eigenschaften der Instanzen → Abschnitt [7.5 auf Seite 71](#)
- Zulässige Operationen → Abschnitt [7.4 auf Seite 69](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [7.6 auf Seite 71](#)

7.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("M")` auf Seite 63 liefert u. a. eine Auflistung der M-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 7.1 aufgeführt.

Tabelle 7.1: Methoden der Klasse M

Name	Bedeutung
<code>M.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse M.
<code>M.classInfo(art)</code>	Gibt Informationen zur Klasse M und ihre Methoden aus.
<code>M.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>M.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse M aus.
<code>m.info(modus)</code>	Gibt Informationen über die M-Instanz <i>m</i> aus.
<code>m.to(u)</code>	Realisiert die Konvertierung der M-Instanz <i>m</i> in eine andere Maßeinheit <i>u</i> .
<code>m.toLongweights()</code>	Liefert <code>lton</code> , <code>cwt</code> , <code>qu</code> , <code>stone</code> , <code>lb</code> , <code>oz</code> , <code>dr</code> , <code>grain</code> der aktuellen M-Instanz <i>m</i> als Liste.

Ausführlichere Informationen über eine bestimmte M-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `M.classInfo("M")` für alle M-Methoden.

Beispiele für dok

```
>>> dok("M.to")
M.to
    Realisiert die Konvertierung einer M-Instanz in eine andere Maßeinheit.

    Aufruf: m.to(u)
    erlaubter Typ des Operanden:
    - eine der zulässigen M-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("M.info")
M.info
    Gibt Informationen über eine M-Instanz aus.

    Aufruf: m.info(modus)
    m Modus
        kurz  nur Grund-Eigenschaften
              [Voreinstellung]
        lang  auch Darstellung in anderen Einheiten
>>> dok("M.toLongweights")
M.toLongweights
```

Liefert lton, cwt, qu, stone, lb, oz, dr, grain der aktuellen Instanz als Liste.

Aufruf: `t.toLongweights()`

7.3 Maßeinheiten und Instanzen

Maßeinheiten

Eine Übersicht über die Maßeinheiten in *M* liefern die globalen Methoden `allUnits("M")` und `au("M")`:

```
>>> allUnits("M")
```

Gewichtsmaße (in kg): 25 Einträge

cental	45.36	amerik. Gewichtsmaß, short Avoirdupois: short hundredweight (Zentner); 100 pound
ct	0.0	Gewichtsmaß: Karat (ct)
cwt	50.8	amerik. Gewichtsmaß, long Avoirdupois: long hundredweight (Zentner); 4 qu
dr	0.0	amerik. Gewichtsmaß, Avoirdupois: dram (Drachme); 1/16 ounce
dz	100	Gewichtsmaß: Doppelzentner (dz)
g	0.0	Gewichtsmaß: Gramm (g)
grain	0.0	amerik. Gewichtsmaß, Avoirdupois: grain; 1/7000 pound
Gt	1000000000000.0	Gewichtsmaß: Gigatonne (Gt)
kg	1	Gewichtsmaß: Kilogramm (kg)
kt	1000000.0	Gewichtsmaß: Kilotonne (kt)
lb	0.45	amerik. Gewichtsmaß, Avoirdupois: pound (Pfund); lb., pd., lbm.
lton	1016.05	amerik. Gewichtsmaß, long Avoirdupois: long ton, T., to.; 20 cwt
mg	0.0	Gewichtsmaß: Milligramm (mg)
Mt	1000000000.0	Gewichtsmaß: Megatonne (Mt)
myg	0.0	Gewichtsmaß: Mikrogramm (µg)
ng	0.0	Gewichtsmaß: Nanogramm (ng)
oz	0.03	amerik. Gewichtsmaß, Avoirdupois: ounce (Unze); 1/16 pound
pfd	0.5	Gewichtsmaß: Pfund (pfd)
pg	0.0	Gewichtsmaß: Pikogramm (pg)
qu	12.7	amerik. Gewichtsmaß, long Avoirdupois: quarter (Viertelzentner); qu., qr. l.; 2 stone
quarter	11.34	amerik. Gewichtsmaß, short Avoirdupois: quarter (Viertelzentner); 1/4 cwt
stone	6.35	amerik. Gewichtsmaß, long Avoirdupois: stone; 14 pound

t	1000.0	Gewichtsmaß: Tonne (t)
ton	907.18	amerik. Gewichtsmaß, short Avoirdupois: short ton; 20 quarter
Ztr	50	Gewichtsmaß: Zentner (Ztr)

Interessant sind die amerikanischen Masseneinheiten/Gewichtsmaße (`cental`, `cwt`, `dr`, `grain`, `lb`, `lton`, `oz`, `qu`, `quarter`, `stone`, `ton`) und die nicht-metrischen Einheiten (`ct`, `pdf`, `Ztr`).

Instanzen

Eigene M-Instanzen können Sie mittels der Anweisung `M(masse, einheit, n=name)` oder der globalen Methoden `kilogramm(wert)` (Gewichtsmaß) bzw. `longweightsToM(liste)` (Gewichtsmaß, amerikanisch) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> m11 = M(2, kg)
>>> m12 = M(-2, mg)
>>> m13 = +M(0.5, t)
>>> m14 = -kilogramm(3.750)      # globale Methode
>>> l   = [0, 0, 0, 0, 1, 2, 3, 4] # [lton, cwt, qu, stone, lb, oz, dr, grain]
>>> m15 = longweightsToM(l)      # globale Methode; l ---> M
>>> m11
M(2,kg)
>>> m12
M(-2,mg)
>>> m13
M(0.5,t)
>>> m14
M(-3.75,kg)
>>> m15
M(0.52,kg)
```

Mögliche Fehlerquellen

Der Versuch, eine eigene M-Instanz zu generieren, kann misslingen:

- `Unit` ist hier nicht bekannt/zulässig: Nur die bei einem Aufruf der globalen Methode `allUnits("M")` genannten Maßeinheiten sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("M.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne M-Instanz erhalten Sie mit der M-Methode `info`, beispielsweise für `m11`:

```
>>> m11.info()
Name       : M(2,kg)
Art        : M-Instanz (Gewicht)
```

```

Wert          : 2 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: 2 (in kg)
-----

```

Informationen über alle M-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```

>>> alle("M")
7 Element(e):

```

```

Name          : M(2,kg)
Art           : M-Instanz (Gewicht)
Wert          : 2 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: 2 (in kg)
-----

```

```

Name          : M(-2,mg)
Art           : M-Instanz (Gewicht)
Wert          : -2 (in mg)
Einheit       : mg [Gewichtsmaß: Milligramm (mg) ]
interner Wert: -0.0 (in kg)
-----

```

```

Name          : M(0.5,t)
Art           : M-Instanz (Gewicht)
Wert          : 0.5 (in t)
Einheit       : t [Gewichtsmaß: Tonne (t) ]
interner Wert: 500.0 (in kg)
-----

```

```

Name          : M(0.5,t)
Art           : M-Instanz (Gewicht)
Wert          : 0.5 (in t)
Einheit       : t [Gewichtsmaß: Tonne (t) ]
interner Wert: 500.0 (in kg)
-----

```

```

Name          : kilogramm(3.75)
Art           : M-Instanz (Gewicht)
Wert          : 3.75 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: 3.75 (in kg)
-----

```

```

Name          : M(-3.75,kg)
Art           : M-Instanz (Gewicht)

```

```

Wert          : -3.75 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: -3.75 (in kg)
-----

```

```

Name          : M(0.52,kg)
Art           : M-Instanz (Gewicht)
Wert          : 0.52 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: 0.52 (in kg)
-----

```

7.4 Operatoren

Vergleiche

Instanzen der Klasse M können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> m11, m12, m13, m14
(M(2,kg), M(-2,mg), M(0.5,t), M(-3.75,kg))
>>> m11 <= m12
False
>>> m13 >= m14
True
>>> m12 != M(23, kg)
True
>>> m14 == M(3.4, t)
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> m11
M(2,kg)
>>> (M(1, kg) <= m11) and (m11 <= M(5, kg))      # m11 zwischen 1kg und 5kg
True
>>> not ((m11 == M(1, kg)) or (m11 == M(5, kg))) # m11 weder 1kg noch 5kg
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse M verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 7.2.

Tabelle 7.2: Klasse M, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	F1	G	G	G	G	G	G	M	G	G	G	G	G	G	M
/	G	G	G	G	G	G	Skalar	M	G	G	G	G	G	G	M
+	–	–	–	–	–	–	–	M	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	M	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```
>>> m11, m12, m13, m14
(M(2,kg), M(-2,mg), M(0.5,t), M(-3.75,kg))
>>> m11 + m12
M(2.0,kg)
>>> m13 - M(23, kg)
M(477.0,kg)
>>> m14 * T1(3.4, h)
G(M(-3.75,kg) * (T1(3.4,h)))
>>> m11 / m12
-1000000.0
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit M-Instanzen erhalten Sie durch die Methode `M.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("M.beispiel")`, beispielsweise `M.beispiel()`.

7.5 Eigenschaften der M-Instanzen

Jede M-Instanz (jedes M-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der M-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `M.classInfo("V")`. Danach stehen in der Klasse M die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> M.classInfo("V")
```

Eigenschaften der M-Instanzen:

```
m.v          Wert der Instanz (gemessen in m.u)
m.u          Maßeinheit der Instanz (M-Maßeinheit)
m.name       Name der Instanz
m.internal   interner Wert
```

Beispiele

```
>>> m14
M(-3.75,kg)
>>> m14.info()
Name          : M(-3.75,kg)
Art           : M-Instanz (Gewicht)
Wert          : -3.75 (in kg)
Einheit       : kg [Gewichtsmaß: Kilogramm (kg)]
interner Wert: -3.75 (in kg)
-----

>>> m14.v
-3.75
>>> m14.u
'kg'
>>> m14.name
'M(-3.75,kg) '
>>> m14.internal
-3.75
```

7.6 Konvertierung

Instanzen der Klasse M können mit der Hilfe der M-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden. Daneben gibt es noch die M-Methode `toLongweights`.

Beispiele – Konvertierung

```

>>> m11, m12, m13, m14
(M(2,kg), M(-2,mg), M(0.5,t), M(-3.75,kg))
>>> m11.to(mg)          # kg --> mg
M(2000000.0,mg)
>>> m12.to(kg)          # mg --> kg
M(-0.0,kg)
>>> m13.to(g)           # t --> g
M(500000.0,g)
>>> m14.to(t)           # kg --> t
M(-0.0,t)
>>> M(2,lb).to(kg)       # lb --> kg
M(0.91,kg)
>>> M(10,kg).to(lb)      # kg --> lb
M(22.05,lb)
>>> M(1.5,lton).to(t)    # lton --> t
M(1.52,t)
>>> mx = m14.toLongweights() # kg --> Longweights
>>> mx[0]; mx[1]; mx[2]; mx[3]; mx[4]; mx[5]; mx[6]
M(0,lton)
M(0,cwt)
M(0,qu)
M(0,stone)
M(-8,lb)
M(-4,oz)
M(-4,dr)
>>> kilogramm(3.5).to(oz) # globale Methode; kg --> oz
M(123.46,oz)
>>> m14.to(s)           # kg --> s ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 6486, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: s

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

8 Basisklasse TT

TT realisiert das Rechnen mit Temperaturmaßen.

8.1 Übersicht

Globale Informationen über die Klasse TT

Eine Übersicht über die Klasse TT erhalten Sie durch die globale Methode `dok("TT")` oder – etwas ausführlicher – mit der TT-Methode `TT.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("TT.classInfo")`:

```
>>> dok("TT.classInfo")
TT.classInfo
    Gibt Informationen zur Klasse TT und ihre Methoden aus.

    Aufruf: TT.classInfo(m=art) oder TT.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen: Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> TT.classInfo("H")
Class TT

    Realisiert das Rechnen mit TT-Instanzen (Temperaturen).

    Aufruf: TT.__init__(value=1, unit=K, n="")

    Parameter:

    value=1: Wert;
              Voreinstellung: 1
    unit=K  : TT-Maßeinheit;
              Voreinstellung: K
    n=""    : Name der TT-Instanz;
              Voreinstellung: leere Zeichenkette

    Methoden:
```

```

tt : TT-Instanz (Temperatur)
o  : Instanz
n  : Zahl(Integer/Float)
p  : Integer
u  : eine der zulässigen TT-Maßeinheiten

```

(alle Operationen beziehen sich auf Grad Kelvin)

```

tt.__add__(tt) oder tt + tt  Realisiert die Addition zweier TT-Instanzen.
tt.__eq__(tt) oder tt == tt  Realisiert den Vergleich == in TT.
tt.__ge__(tt) oder tt >= tt  Realisiert den Vergleich >= in TT.
tt.__gt__(tt) oder tt > tt   Realisiert den Vergleich > in TT.
TT.__init__(value=1,unit=K,n="") Initialisiert eine TT-Instanz.
tt.__le__(tt) oder tt <= tt  Realisiert den Vergleich <= in TT.
tt.__lt__(tt) oder tt < tt   Realisiert den Vergleich < in TT.
tt.__mul__(o) oder tt * o    Realisiert die Multiplikation in TT.
tt.__ne__(tt) oder tt != tt  Realisiert den Vergleich != in TT.
tt.__neg__() oder -(tt)      Realisiert negatives Vorzeichen für eine
                             TT-Instanz. (nicht verfügbar)
tt.__pos__() oder +(tt)      Realisiert positives Vorzeichen für eine
                             TT-Instanz. (nicht verfügbar)
tt.__pow__(p) oder tt ** p    Realisiert das Potenzieren einer TT-Instanz.
tt.__repr__() oder repr(tt)   Repräsentiert eine TT-Instanz.
tt.__str__() oder str(tt)     Repräsentiert eine TT-Instanz.
tt.__sub__(tt) oder tt - tt   Realisiert die Subtraktion zweier TT-Instanzen.
tt.__truediv__(o) oder tt / o Realisiert die Division einer TT-Instanz.
TT.beispiel(e)                Liefert Beispiele für die Methoden der Klasse TT.
TT.classInfo(m=art)           Gibt Informationen zur Klasse TT und ihre
                             Methoden aus.
TT.ci(m=art)                  Alias für TT.classInfo(m=art)
TT.description()              Gibt eine Kurzbeschreibung der Klasse TT aus.
tt.info(modus)                Gibt Informationen über eine TT-Instanz aus.
tt.to(u)                      Realisiert die Konvertierung einer TT-Instanz in
                             eine andere Maßeinheit.
-----

```

Weitere Informationen

- Methoden der Klasse → Abschnitt 8.2 auf der nächsten Seite
- Maßeinheiten und Instanzen → Abschnitt 8.3 auf Seite 76
- Eigenschaften der Instanzen → Abschnitt 8.5 auf Seite 79
- Zulässige Operationen → Abschnitt 8.4 auf Seite 77
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 8.6 auf Seite 80

8.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("TT")` auf Seite 73 liefert u. a. eine Auflistung der TT-Methoden. Namen, die die Zeichenfolge `--` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 8.1 aufgeführt.

Tabelle 8.1: Methoden der Klasse TT

Name	Bedeutung
<code>TT.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse TT.
<code>TT.classInfo(art)</code>	Gibt Informationen zur Klasse TT und ihre Methoden aus.
<code>TT.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>TT.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse TT aus.
<code>t.info(modus)</code>	Gibt Informationen über die TT-Instanz <i>t</i> aus.
<code>t.to(u)</code>	Realisiert die Konvertierung der TT-Instanz <i>t</i> in eine andere Maßeinheit <i>u</i> .

Ausführlichere Informationen über eine bestimmte TT-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `TT.classInfo("M")` für alle TT-Methoden.

Beispiele für dok

```
>>> dok("TT.to")
```

```
TT.to
```

```
Realisiert die Konvertierung einer TT-Einheit in eine andere TT-Maßeinheit.
```

```
Aufruf: tt.to(unit)
```

```
erlaubter Typ des Operanden:
```

```
- eine der zulässigen TT-Maßeinheiten
```

```
mögliche Fehlermeldung:
```

```
- Unit ist hier nicht bekannt/zulässig
```

```
>>> dok("TT.info")
```

```
TT.info
```

```
Gibt Informationen über eine TT-Instanz aus.
```

```
Aufruf: tt.info(m)
```

```
m Modus
```

```
kurz nur Grund-Eigenschaften
```

```
[Voreinstellung]
```

```
lang auch Darstellung in anderen Einheiten
```

8.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse TT stellt die Maßeinheiten C, R, F und K zur Verfügung.

Instanzen

Eigene TT-Instanzen können Sie mittels der Anweisung `TT(grad, einheit, n=name)` oder der globalen Methode `kelvin(wert)` generieren:

```
>>> tt1 = TT(2, C)
>>> tt2 = TT(-2, R)
>>> tt3 = TT(0.5, F)
>>> tt4 = kelvin(3.750) # globale Methode
>>> tt1, tt2, tt3, tt4
(TT(2,C), TT(-2,R), TT(0.5,F), TT(3.75,K))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene TT-Instanz zu generieren, kann misslingen:

- Unit ist hier nicht bekannt/zulässig: Nur die Maßeinheiten C, R, F und K sind zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("TT.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne TT-Instanz erhalten Sie mit der TT-Methode `info`, beispielsweise für `tt4`:

```
>>> tt4.info()
Name          : kelvin(3.75)
Art           : TT-Instanz (Temperatur)
Wert          : 3.75 (in K)
Einheit       : K [Temperaturmaß: Kelvin]
interner Wert: 3.75 (in K)
-----
```

Informationen über alle TT-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("TT")
4 Element(e):

Name          : TT(2,C)
Art           : TT-Instanz (Temperatur)
Wert          : 2 (in C)
Einheit       : C [Temperaturmaß: Celsius]
interner Wert: 275.15 (in K)
```

```

-----
Name      : TT(-2,R)
Art       : TT-Instanz (Temperatur)
Wert      : -2 (in R)
Einheit   : R [Temperaturmaß: Reaumur]
interner Wert: 270.65 (in K)
-----

```

```

-----
Name      : TT(0.5,F)
Art       : TT-Instanz (Temperatur)
Wert      : 0.5 (in F)
Einheit   : F [Temperaturmaß: Fahrenheit]
interner Wert: 255.65 (in K)
-----

```

```

-----
Name      : kelvin(3.75)
Art       : TT-Instanz (Temperatur)
Wert      : 3.75 (in K)
Einheit   : K [Temperaturmaß: Kelvin]
interner Wert: 3.75 (in K)
-----

```

8.4 Operatoren

Vergleiche

Instanzen der Klasse TT können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> tt1, tt2, tt4
(TT(2,C), TT(-2,R), TT(3.75,K))
>>> tt1 == tt2
False
>>> tt1 <= tt4
False
>>> TT(2, K) != TT(3, K)
True

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> tt4
TT(3.75,K)
>>> (TT(100, K) <= tt4) and (tt4 <= TT(300, K))      # tt4 zwischen 100K und 300K
False
>>> not ((tt4 == TT(100, K)) or (tt4 == TT(300, K))) # tt4 weder 100K noch 300K
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse TT verfügt über keine arithmetischen Operatoren. Siehe dazu Tabelle 8.2.

Tabelle 8.2: Klasse TT, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(TT)	(G)	(G)	(TT)
/	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(Skalar)	(G)	(G)	(TT)
+	–	–	–	–	–	–	–	–	–	–	–	(TT)	–	–	–
–	–	–	–	–	–	–	–	–	–	–	–	(TT)	–	–	–
**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Beispiele – arithmetische Operatoren

```
>>> tt1, tt2, tt3, tt4
(TT(2,C), TT(-2,R), TT(0.5,F), TT(3.75,K))
>>> tt1 + tt2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 8615, in __add__
    raise ValueError(_fehler2 + str(tt.u))
ValueError: Unit ist hier nicht bekannt/zulässig: R
>>> tt2 - tt4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 8871, in __sub__
    raise ValueError(_fehler2 + str(tt.u))
ValueError: Unit ist hier nicht bekannt/zulässig: K
>>> tt4 + TT(100, K)
TT(103.75,K)
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit TT-Instanzen erhalten Sie durch den Aufruf der TT-Methode `TT.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("TT.beispiel")`, beispielsweise `TT.beispiel()`.

8.5 Eigenschaften der TT-Instanzen

Jede TT-Instanz (jedes TT-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der TT-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `TT.classInfo("V")`. Danach stehen in der Klasse TT die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> TT.classInfo("V")
```

Eigenschaften der TT-Instanzen:

```
t.v          Wert der Instanz (gemessen in t.u)
t.u          Maßeinheit der Instanz (TT-Maßeinheit)
t.name       Name der Instanz
t.internal   interner Wert
```

Beispiele

```
>>> tt3
```

```
TT(0.5,F)
```

```
>>> tt3.info()
```

```
Name          : TT(0.5,F)
Art           : TT-Instanz (Temperatur)
Wert          : 0.5 (in F)
Einheit       : F [Temperaturmaß: Fahrenheit]
interner Wert: 255.65 (in K)
```

```
-----
```

```
>>> tt3.v; tt3.u; tt3.name; tt3.internal
```

```
0.5
```

```
'F'
```

```
'TT(0.5,F)'
```

```
255.64999999999998
```

8.6 Konvertierung

Instanzen der Klasse TT können mit der Hilfe der TT-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden:

Beispiele – Konvertierung

```
>>> tt1, tt2, tt3, tt4
(TT(2,C), TT(-2,R), TT(0.5,F), TT(3.75,K))
>>> tt1.to(F)          # C --> F
TT(35.6,F)
>>> tt2.to(F)          # R --> F
TT(27.5,F)
>>> tt3.to(K)          # F --> K
TT(255.65,K)
>>> tt4.to(C)          # K --> C
TT(-269.4,C)
>>> kelvin(239).to(F)  # K --> F # globale Methode
TT(-29.47,F)
>>> tt4.to(m)          # K --> m ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 9080, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: m
```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

9 Basisklasse G

9.1 Übersicht

Globale Informationen über die Klasse G

Eine Übersicht über die Klasse G erhalten Sie durch die globale Methode `dok("G")` oder – etwas ausführlicher – mit der G-Methode `G.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("G.classInfo")`:

```
>>> dok("G.classInfo")
G.classInfo
    Gibt Informationen zur Klasse G und ihre Methoden aus.

    Aufruf: G.classInfo(m=art) oder G.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> G.classInfo("H")
Class G

    Realisiert das Kombinieren von Instanzen (int, float, L1, L2, L3, M, T1, T2,
    TT, N) mittels eines Operators (*,/).

    g.__eq__(g)                Vergleicht die Maßeinheiten zweier G-Instanzen.
                                (intern)
    g.__add__(g, k=konvertierung) oder g + g
                                Dient der Addition von G-Instanzen.
    g.__eq__(g) oder g == g    Realisiert den Vergleich == in G.
    g.__ge__(g) oder g >= g    Realisiert den Vergleich >= in G.
    g.__gt__(g) oder g > g     Realisiert den Vergleich > in G.
    G.__init__(l, op, r, n='') Initialisiert eine G-Instanz.
    g.__le__(g) oder g <= g    Realisiert den Vergleich <= in G.
    g.__lt__(g) oder g < g     Realisiert den Vergleich < in G.
    g.__mul__(o) oder g * o    Realisiert die Multiplikation von G-Instanzen.
    g.__ne__(g) oder g != g    Realisiert den Vergleich != in G.
    g.__neg__() oder -(g)      Realisiert negatives Vorzeichen von G-Instanzen.
    g.__pos__() oder +(g)      Realisiert positives Vorzeichen von G-Instanzen.
```

<code>g.__repr__() oder repr(g)</code>	Repräsentiert eine <i>G</i> -Instanz.
<code>g.__str__() oder str(g)</code>	Repräsentiert eine <i>G</i> -Instanz.
<code>g.__sub__(g, k=konvertierung) oder g - g</code>	Dient der Subtraktion von <i>G</i> -Instanzen.
<code>g.__truediv__(o) oder g / o</code>	Realisiert die Division von <i>G</i> -Instanzen.
<code>G.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse <i>G</i> .
<code>G.classInfo(m=art)</code>	Gibt Informationen zur Klasse <i>G</i> und ihre Methoden aus.
<code>G.ci(m=art)</code>	Alias für <code>G.classInfo(m=art)</code>
<code>G.description()</code>	Liefert eine Kurzbeschreibung der Klasse <i>G</i> .
<code>g.info()</code>	Gibt Informationen über eine <i>G</i> -Instanz aus.
<code>g.to(liste)</code>	Gibt in geeigneter Weise den Wert einer <i>G</i> -Instanz aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt 9.2
- Maßeinheiten und Instanzen → Abschnitt 9.3 auf der nächsten Seite
- Eigenschaften der Instanzen → Abschnitt 9.5 auf Seite 89
- Zulässige Operationen → Abschnitt 9.4 auf Seite 87
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt 9.6 auf Seite 90

9.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("G")` auf der vorherigen Seite liefert u. a. eine Auflistung der *G*-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 9.1 aufgeführt.

Tabelle 9.1: Methoden der Klasse *G*

Name	Bedeutung
<code>G.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse <i>G</i> .
<code>G.classInfo(art)</code>	Gibt Informationen zur Klasse <i>G</i> und ihre Methoden aus.
<code>G.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>G.description()</code>	Liefert eine einzeilige Kurzbeschreibung der Klasse <i>G</i> .
<code>g.info(modus)</code>	Gibt Informationen über die <i>G</i> -Instanz <i>g</i> aus.
<code>g.to(liste)</code>	Gibt in geeigneter Weise den Wert der <i>G</i> -Instanz <i>g</i> aus.

Ausführlichere Informationen über eine bestimmte *G*-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `G.classInfo("M")` für alle *G*-Methoden.

Beispiele für dok

```
>>> dok("G.to")
G.to
    Gibt in geeigneter Weise den Wert einer G-Instanz aus.

    Aufruf: g.to(wliste)
    vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
    - leer : keine Konvertierung [Voreinstellung], aber Bereinigung
    - kmh  : m --> km, s --> h
    - mks  : m --> m, s --> s, kg --> kg
    - cgs  : m --> cm, kg --> g, s --> s
    - mph  : m --> mi, s --> h
    mögliche Fehlermeldung:
    - Parameter ist unzulässig.

>>> dok("G.info")
G.info
    Gibt Informationen über eine G-Instanz aus.

    Aufruf: g.info()
```

9.3 Maßeinheiten und Instanzen**Maßeinheiten**

Die Klasse G besitzt keine eigenen Maßeinheiten.

Instanzen

Eigene G-Instanzen können Sie mittels der Anweisung `G(links, operator, rechts, n=name)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> l1 = L1(3, m)
>>> t1 = T1(2, s)
>>> t2 = T2(1.5, s2)
>>> m0 = M(1.7, kg)
>>> v0 = meterS(3) # globale Methode
>>> l1, t1, t2, m0, v0
(L1(3,m), T1(2,s), T2(1.5,s2), M(1.7,kg), V(L1(3,m), T1(1,s)))
>>> g1 = G(l1, div, t1) # Geschwindigkeit
>>> g2 = G(l1, div, t2) # Beschleunigung
>>> g3 = -G(t1, mul, v0, n="Weg") # Strecke
>>> g4 = +G(m0, mul, v0, n="Impuls") # Impuls
>>> g1
G(L1(3,m) / (T1(2,s)))
>>> g2
G(L1(3,m) / (T2(1.5,s2)))
```

```
>>> g3
G(T1(-2,s) * (V(L1(3,m), T1(1,s))))
>>> g4
G(M(1.7,kg) * (V(L1(3,m), T1(1,s))))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene G-Instanz zu generieren, kann misslingen:

- **Operand hat den Wert Null:** Der rechte Operand *rechts* hat den Wert Null (wenn der Operator `div` ist).
- **Operand hat keinen zulässigen Typ:** Für den linken *links* bzw. rechten Operanden *rechts* sind nur die Typen `int`, `float`, `L1`, `L2`, `L3`, `T1`, `T2`, `M`, `TT`, `G`, `N` zulässig.
- **Operator hat keinen zulässigen Typ:** Für *operator* sind nur `mul` und `div` zulässig.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("G.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne G-Instanz erhalten Sie mit der G-Methode `info`, beispielsweise für `g1`:

```
>>> g1.info()
Name           : G(L1(3,m) / (T1(2,s)))
Art            : G-Instanz
linker Operand : L1(3,m)
Operator       : /
rechter Operand: T1(2,s)
Wert           : 1.5 (in den angegebenen Maßeinheiten)
interner Wert  : 1.5 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
-----
```

Informationen über alle G-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("G")
10 Element(e):

Name           : L2(3.75,m2) * (L2(3.4,km2))
Art            : G-Instanz
linker Operand : L2(3.75,m2)
Operator       : *
rechter Operand: L2(3.4,km2)
Wert           : 12.75 (in den angegebenen Maßeinheiten)
interner Wert  : 12750000.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m2', 'km2']
Maßeinheit(en) Nenner: []
-----
```

9 Basisklasse G

```

Name          : L3(3.75,m3) * (L3(3.4,cm3))
Art           : G-Instanz
linker Operand : L3(3.75,m3)
Operator      : *
rechter Operand : L3(3.4,cm3)
Wert          : 12.75 (in den angegebenen Maßeinheiten)
interner Wert  : 0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m3', 'cm3']
Maßeinheit(en) Nenner: []
-----

Name          : T2(3.75,s2) * (T2(3.4,h2))
Art           : G-Instanz
linker Operand : T2(3.75,s2)
Operator      : *
rechter Operand : T2(3.4,h2)
Wert          : 12.75 (in den angegebenen Maßeinheiten)
interner Wert  : 165240000.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['s2', 'h2']
Maßeinheit(en) Nenner: []
-----

Name          : M(-3.75,kg) * (T1(3.4,h))
Art           : G-Instanz
linker Operand : M(-3.75,kg)
Operator      : *
rechter Operand : T1(3.4,h)
Wert          : -12.75 (in den angegebenen Maßeinheiten)
interner Wert  : -45900.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'h']
Maßeinheit(en) Nenner: []
-----

Name          : meterS(3)
Art           : V-Instanz (Geschwindigkeit)
linker Operand : L1(3,m)
Operator      : /
rechter Operand : T1(1,s)
Wert          : 3.0 (in den angegebenen Maßeinheiten)
interner Wert  : 3.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 10.8 ['km']/['h']
konvertiert (cgs)  : 300.0 ['cm']/['s']
konvertiert (mks)  : 3.0 ['m']/['s']
-----

```

9 Basisklasse G

```
Name          : G(L1(3,m) / (T1(2,s)))
Art           : G-Instanz
linker Operand : L1(3,m)
Operator      : /
rechter Operand : T1(2,s)
Wert          : 1.5 (in den angegebenen Maßeinheiten)
interner Wert  : 1.5 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
```

```
Name          : G(L1(3,m) / (T2(1.5,s2)))
Art           : G-Instanz
linker Operand : L1(3,m)
Operator      : /
rechter Operand : T2(1.5,s2)
Wert          : 2.0 (in den angegebenen Maßeinheiten)
interner Wert  : 2.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s2']
```

```
Name          : Weg
Art           : G-Instanz
linker Operand : T1(2,s)
Operator      : *
rechter Operand : V(L1(3,m), T1(1,s))
Wert          : 6.0 (in den angegebenen Maßeinheiten)
interner Wert  : 6.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['s', 'm']
Maßeinheit(en) Nenner: ['s']
```

```
Name          : G(T1(-2,s) * (V(L1(3,m), T1(1,s))))
Art           : G-Instanz
linker Operand : T1(-2,s)
Operator      : *
rechter Operand : V(L1(3,m), T1(1,s))
Wert          : -6.0 (in den angegebenen Maßeinheiten)
interner Wert  : -6.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['s', 'm']
Maßeinheit(en) Nenner: ['s']
```

```
Name          : Impuls
Art           : G-Instanz
```

```

linker Operand      : M(1.7,kg)
Operator            : *
rechter Operand     : V(L1(3,m), T1(1,s))
Wert                : 5.1 (in den angegebenen Maßeinheiten)
interner Wert       : 5.1 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'm']
Maßeinheit(en) Nenner: ['s']
-----

```

9.4 Operatoren

Vergleiche

Instanzen der Klasse **G** können mit Einschränkungen mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> g1; g2; g3; g4
G(L1(3,m) / (T1(2,s)))
G(L1(3,m) / (T2(1.5,s2)))
G(T1(-2,s) * (V(L1(3,m), T1(1,s))))
G(M(1.7,kg) * (V(L1(3,m), T1(1,s))))
>>> g1 <= g2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: '<=' not supported between instances of 'G' and 'G'
>>> g3 >= g4
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: '>=' not supported between instances of 'G' and 'G'
>>> g2 != g1
True
>>> g1 == g4
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> g1; g2; g3
G(L1(3,m) / (T1(2,s)))
G(L1(3,m) / (T2(1.5,s2)))
G(T1(-2,s) * (V(L1(3,m), T1(1,s))))
>>> not ((g1 == g2) or (g1 == g3)) # g1 weder g2 noch g3
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse G verfügt nur über die arithmetischen Operatoren * und /. Siehe dazu Tabelle 9.2.

Tabelle 9.2: Klasse G, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
/	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
+	–	–	(G)	–	–	–	–	–	–	–	–	–	–	–	–
–	–	–	(G)	–	–	–	–	–	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Beispiele

```
>>> g1; g2
G(L1(3,m) / (T1(2,s)))
G(L1(3,m) / (T2(1.5,s2)))
>>> g1 * g2
G(G(L1(3,m) / (T1(2,s))) * (G(L1(3,m) / (T2(1.5,s2)))))
>>> g1 * T1(2, s)
G(G(L1(3,m) / (T1(2,s))) * (T1(2,s)))
>>> g2 * L1(23, m)
G(G(L1(3,m) / (T2(1.5,s2))) * (L1(23,m)))
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit G-Instanzen erhalten Sie durch die Methode `G.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("G.beispiel")`, beispielsweise `G.beispiel()`.

9.5 Eigenschaften der G-Instanzen

Jede G-Instanz (jedes G-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der G-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `G.classInfo("V")`. Danach stehen in der Klasse G die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> G.classInfo("V")
```

Eigenschaften der G-Instanzen:

```
g.links      linker Operand
g.operator    Operator
g.rechts      rechter Operand
g.name        Name der Instanz
g.oben        Einheiten im Zähler
g.unten       Einheiten im Nenner
g.internal    interner Wert
g.v           Wert in den angegebenen Einheiten
```

Beispiele

```
>>> g4
```

```
G(M(1.7,kg) * (V(L1(3,m), T1(1,s))))
```

```
>>> g4.info()
```

```
Name           : Impuls
Art             : G-Instanz
linker Operand  : M(1.7,kg)
Operator        : *
rechter Operand : V(L1(3,m), T1(1,s))
Wert            : 5.1 (in den angegebenen Maßeinheiten)
interner Wert   : 5.1 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'm']
Maßeinheit(en) Nenner: ['s']
```

```
>>> g4.links
```

```
M(1.7,kg)
```

```
>>> g4.operator
```

```
'*'
```

```
>>> g4.rechts
```

```
V(L1(3,m), T1(1,s))
```

```
>>> g4.name
```

```
'Impuls'
```

```
>>> g4.oben
```

```
['kg', 'm']
```

```
>>> g4.unten
['s']
>>> g4.internal
5.1
>>> g4.v
5.1
```

9.6 Konvertierung

Instanzen der Klasse G können mit der Hilfe der G-Methode (*einheitensystem*) in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

leer : keine Konvertierung, aber Vereinigung

kmh : $m \rightarrow km, s \rightarrow h$

mks : $m \rightarrow m, s \rightarrow s, kg \rightarrow kg$

cgs : $m \rightarrow cm, kg \rightarrow g, s \rightarrow s$

mph : $m \rightarrow mi, s \rightarrow h$

Beispiele – Konvertierung

```
>>> g1; g2; g3; g4
G(L1(3,m) / (T1(2,s)))
G(L1(3,m) / (T2(1.5,s2)))
G(T1(-2,s) * (V(L1(3,m), T1(1,s))))
G(M(1.7,kg) * (V(L1(3,m), T1(1,s))))
>>> g1.to(kmh)
"5.4 ['km']/['h']"
>>> g2.to(mph)
"16105.94 ['mi']/['h', 'h']"
>>> g3.to(cgs)
L1(-600.0,cm)
>>> g4.to()
"5.1 ['kg', 'm']/['s']"
>>> g4.to(cgs)
"510000.0 ['g', 'cm']/['s']"
>>> g4.to(mkh)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'mkh' is not defined
```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

10 Basisklasse N

N ist eine Hilfsklasse. Sie kann sinnvoll eingesetzt werden, wenn eine Instanz linksseitig mit einem Skalar multipliziert werden soll.

10.1 Übersicht

Globale Informationen über die Klasse N

Eine Übersicht über die Klasse N erhalten Sie durch die globale Methode `dok("N")` oder – etwas ausführlicher – mit der N-Methode `N.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("N.classInfo")`:

```
>>> dok("N.classInfo")
N.classInfo
    Gibt Informationen zur Klasse N und ihre Methoden aus.

    Aufruf: N.classInfo(m=art) oder N.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> N.classInfo("H")
Class N

    Realisiert den linksseitigen Operator (*, /) für (int, float, L1, L2, L3, M,
    T1, T2, TT, N, U, V, B, F1, W, P).

    Aufruf: N.__init__(value=1, n='')

    n : N-Instanz
    o : Instanz
    p : Zahl (int, Float)

    n.__abs__() oder abs(n)    Liefert den Absolut-Betrag der N-Instanz.
    n.__add__(n) oder n + n    Realisiert die Addition zweier N-Instanzen in N.
    n.__eq__(n)                Realisiert den Vergleich == in N.
    n.__ge__(n)                Realisiert den Vergleich >= in N.
```

<code>n.__gt__(n)</code>	Realisiert den Vergleich <code>></code> in N.
<code>N.__init__(value, n='')</code>	Initialisiert eine N-Instanz in N.
<code>n.__le__(n)</code>	Realisiert den Vergleich <code><=</code> in N.
<code>n.__lt__(n)</code>	Realisiert den Vergleich <code><</code> in N.
<code>n.__mul__(o)</code> oder <code>n * o</code>	Realisiert die Multiplikation mit einem Operanden in N.
<code>n.__ne__(n)</code>	Realisiert den Vergleich <code>!=</code> in N.
<code>n.__neg__(n)</code> oder <code>-(n)</code>	Realisiert negatives Vorzeichen für eine N-Instanz.
<code>n.__pos__(n)</code> oder <code>+(n)</code>	Realisiert positives Vorzeichen für eine N-Instanz.
<code>n.__pow__(p)</code> oder <code>t ** p</code>	Realisiert das Potenzieren von N-Instanzen.
<code>n.__repr__()</code> oder <code>repr(n)</code>	Repräsentiert eine N-Instanz.
<code>n.__str__()</code> oder <code>str(n)</code>	Repräsentiert eine N-Instanz.
<code>n.__sub__(n)</code> oder <code>n - n</code>	Realisiert die Subtraktion zweier N-Instanzen
<code>n.__truediv__(o)</code> oder <code>n / o</code>	Realisiert die Division durch einen Operanden in N.
<code>N.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse N.
<code>N.classInfo(m=art)</code>	Gibt Informationen zur Klasse N und ihre Methoden aus.
<code>N.ci(m=art)</code>	Alias für <code>N.classInfo(m=art)</code>
<code>N.description()</code>	Kurzbeschreibung der Klasse N.
<code>n.info()</code>	Gibt Informationen über eine N-Instanz aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt 10.2
- Maßeinheiten und Instanzen → Abschnitt 10.3 auf der nächsten Seite
- Eigenschaften der Instanzen → Abschnitt 10.5 auf Seite 96
- Zulässige Operationen → Abschnitt 10.4 auf Seite 94

10.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("N")` auf der vorherigen Seite liefert u. a. eine Auflistung der N-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 10.1 aufgeführt.

Tabelle 10.1: Methoden der Klasse N

Name	Bedeutung
<code>N.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse N.
<code>N.classInfo(art)</code>	Gibt Informationen zur Klasse N und ihre Methoden aus.
<code>N.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>N.description()</code>	Kurzbeschreibung der Klasse N.
<code>n.info(modus)</code>	Gibt Informationen über der N-Instanz <i>n</i> aus.

Ausführlichere Informationen über eine bestimmte N-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `N.classInfo("M")` für alle N-Methoden.

Beispiel für dok

```
>>> dok("N.info")
N.info
    Gibt Informationen über eine N-Instanz aus.

    Aufruf: n.info()
```

10.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse N besitzt keine eigenen Maßeinheiten.

Instanzen

Eigene N-Instanzen können Sie mittels der Anweisung `N(wert, n=name)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> n1 = N(0.5, n="halbieren")
>>> n2 = -N(2.0, "verdoppeln")
>>> n1
N(0.5)
>>> n2
N(-2.0)
```

Mögliche Fehlerquellen

Für die Klasse N gibt es keine spezifischen Fehlermeldungen.

Informationen über Instanzen

Informationen über eine einzelne N-Instanz erhalten Sie mit der N-Methode `info`, beispielsweise für `n1`:

```
>>> n1.info()
Name          : halbieren
Art           : N-Instanz
interner Wert: 0.5
-----
```

Informationen über alle N-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("N")
3 Element(e):

Name          : halbieren
Art           : N-Instanz
interner Wert: 0.5
-----

Name          : verdoppeln
Art           : N-Instanz
interner Wert: 2.0
-----

Name          : N(-2.0)
Art           : N-Instanz
interner Wert: -2.0
-----
```

10.4 Operatoren

Vergleiche

Instanzen der Klasse N können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele

```
>>> n1, n2
(N(0.5), N(-2.0))
>>> n1 <= n2
False
>>> n2 >= N(14)
False
>>> n1 != N(23)
True
```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> n1
N(0.5)
>>> (N(1) <= n1) and (n1 <= N(5))      # n1 zwischen 1 und 5
False
>>> not ((n1 == N(1)) or (n1 == N(5))) # n1 weder 1 noch 5
True

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse N verfügt über die arithmetischen Operatoren +, -, *, / und **. Siehe dazu Tabelle 10.2.

Tabelle 10.2: Klasse N, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	N
/	G	G	G	G	G	G	G	N	G	G	G	G	G	G	N
+	–	–	–	–	–	–	–	N	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	N	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	N	–	–	–	–	–	–	N

Beispiele – arithmetische Operatoren

```

>>> n1, n2
(N(0.5), N(-2.0))
>>> n1 + n2
N(-1.5)
>>> n2 - N(45)
N(-47.0)
>>> n2 * T1(3.4, h)
T1(-6.8, h)
>>> N(4) * L1(2, m)
L1(8, m)
>>> n2 ** 3
N(-8.0)

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit N-Instanzen erhalten Sie durch die Methode `N.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("N.beispiel")`, hier beispielsweise `N.beispiel()`.

10.5 Eigenschaften der N-Instanzen

Jede N-Instanz (jedes N-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der N-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `N.classInfo("V")`. Danach stehen in der Klasse N die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> N.classInfo("V")
```

Eigenschaften der N-Instanzen:

```

n.v          Wert der Instanz
n.internal   interner Wert
n.name       Name der Instanz

```

Beispiele

```
>>> n2.info()
```

```

Name          : N(-2.0)
Art           : N-Instanz
interner Wert : -2.0

```

```
-----
```

```
>>> n2.v
```

```
-2.0
```

```
>>> n2.name
```

```
'N(-2.0)'
```

```
>>> n2.internal
```

```
-2.0
```

11 Abgeleitete Klasse U

U ist eine Subklasse von T1 und erbt daher auch alle Eigenschaften (Abschnitt 5.5 auf Seite 52) und Methoden (Abschnitt 5.2 auf Seite 47) von T1 (Abschnitt 5 auf Seite 45). Sie dient zum Rechnen mit Uhrzeiten.

11.1 Übersicht

Globale Informationen über die Klasse U

Eine Übersicht über die Klasse U erhalten Sie durch die globale Methode `dok(Ü)` oder – etwas ausführlicher – mit der U-Methode `U.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok(Ü.classInfo)`:

```
>>> dok("U.classInfo")
U.classInfo
    Gibt Informationen zur Klasse U und ihre Methoden aus.

    Aufruf: U.classInfo(m=art) oder U.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> U.classInfo("H")
Class U

    U ist eine Subklasse von T1 und realisiert das Rechnen mit U-Instanzen
    (Uhrzeiten). Sie erbt damit alle Eigenschaften und Methoden von T1
    (außer denen, die überschrieben werden.

    Aufruf: U.__init__(v1=0, v2=0, V3=0, n="")

    Parameter:

    v1=0 : Wert der Stunden;   Voreinstellung: 0
    v2=0 : Wert der Minuten;   Voreinstellung: 0
    v3=0 : Wert der Sekunden;  Voreinstellung: 0
    n=""  : Name der U-Instanz; Voreinstellung: leere Zeichenkette
```

Methoden:

```

u : U-Instanz (Uhrzeit)
e : eine Methode der Klasse U
m : eine der zulässigen U-Maßeinheiten
o : c

u.__abs__() oder abs(u)      Liefert den Absolut-Betrag der U-Instanz.
u.__add__(u) oder u + u     Realisiert die Addition zweier U-Instanzen.
u.__eq__(u) oder u == u     Realisiert den Vergleich == in U.
u.__ge__(u) oder u >= u     Realisiert den Vergleich >= in U.
u.__gt__(u) oder u > u      Realisiert den Vergleich > in U.
U.__init__(v1=0, v2=0, v3=0, n="") Initialisiert eine U-Instanz.
u.__le__(u) oder u <= u     Realisiert den Vergleich <= in U.
u.__lt__(u) oder u < u      Realisiert den Vergleich < in U.
u.__mul__(o) oder u * o     Realisiert die Multiplikation einer U-Instanz.
u.__ne__(u) oder u != u     Realisiert den Vergleich != in U.
u.__neg__() oder -(u)       Realisiert negatives Vorzeichen in U.
u.__pos__() oder +(u)       Realisiert positives Vorzeichen in U.
u.__repr__() oder repr(u)   Repräsentiert eine U-Instanz.
u.__str__() oder str(u)     Repräsentiert eine U-Instanz.
u.__sub__(u) oder u - u     Realisiert die Subtraktion in U.
u.__truediv__(o) oder u / o Realisiert die Division einer U-Instanz.
U.beispiel(e)               Liefert Beispiele für die Methoden der Klasse U.
U.classInfo(m=art)          Gibt Informationen zur Klasse U und ihre Methoden
                             aus.
U.ci(m=art)                 Alias für U.classInfo(m=art)
U.description()             Gibt eine Kurzbeschreibung der Klasse U aus.
u.info(modus)               Gibt Informationen über eine U-Instanz aus.
u.to(m)                     Realisiert die Konvertierung einer U-Instanz in
                             eine andere Maßeinheit.
u.toYMD()                   Liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde
                             der aktuellen U-Instanz als Liste.
-----

```

Weitere Informationen

- Methoden der Klasse → Abschnitt [11.2 auf der nächsten Seite](#)
- Maßeinheiten und Instanzen → Abschnitt [11.3 auf Seite 100](#)
- Eigenschaften der Instanzen → Abschnitt [11.5 auf Seite 103](#)
- Zulässige Operationen → Abschnitt [11.4 auf Seite 101](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [11.6 auf Seite 104](#)

11.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok(Ü)` auf Seite 97 liefert u. a. eine Auflistung der U-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 11.1 aufgeführt.

Tabelle 11.1: Methoden der Klasse U

Name	Bedeutung
<code>U.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse U.
<code>U.classInfo(art)</code>	Gibt Informationen zur Klasse U und ihre Methoden aus.
<code>U.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>U.description()</code>	Gibt eine einzeilige Kurzbeschreibung der Klasse U aus.
<code>u.info(modus)</code>	Gibt Informationen über die U-Instanz <i>u</i> aus.
<code>u.to(u)</code>	Realisiert die Konvertierung der U-Instanz <i>u</i> in eine andere kompatible Maßeinheit <i>u</i> .
<code>u.toYMD()</code>	Liefert Jahr (Year), Monat (Month), Tag (Day), Stunde, Minute, Sekunde der aktuellen U-Instanz <i>u</i> als Liste.

Ausführlichere Informationen über eine bestimmte U-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `U.classInfo("M")` für alle U-Methoden.

Beispiele für dok

```
>>> dok("U.to")
U.to
    Realisiert die Konvertierung einer T1-Instanz in eine andere Maßeinheit.

    Aufruf: t.to(unit)
    erlaubter Typ des Operanden:
    - eine der zulässigen T1-Maßeinheiten
    mögliche Fehlermeldung:
    - Unit ist hier nicht bekannt/zulässig
>>> dok("U.info")
U.info
    Gibt Informationen über eine U-Instanz aus.

    Aufruf: u.info(modus)
    mögliche Angaben für m:
    - kurz  nur Grundangaben
           [Voreinstellung]
    - lang  zusätzlich auch Darstellung in anderen Einheiten
>>> dok("U.toYMD")
U.toYMD
```

Liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde der aktuellen Instanz als Liste.

Aufruf: `t.toYMD()`

11.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse U besitzt keine eigenen Maßeinheiten; sie nutzt die von T1 (Abschnitt 5.3 auf Seite 48).

Instanzen

Eigene U-Instanzen können Sie mittels der Anweisung `U(stunde, minute, sekunde, n=name)` generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> u1 = U(10, 11, 12)
>>> u2 = -U(20, 21, 22)
>>> u1
10:11:12
>>> u2
T1(-73282,s)
```

Mögliche Fehlerquellen

Der Versuch, eine eigene U-Instanz zu generieren, kann misslingen:

- **Parameter ist unzulässig:** Die folgenden Bedingungen müssen eingehalten werden: $-24 \leq stunde \leq 24$; $-60 \leq minute < 60$; $-60 \leq sekunde < 60$
- **Operand hat keinen zulässigen Typ:** Die ersten zwei Parameter (*stunde* und *minute*) müssen vom Typ `int` sein, der dritte *sekunde* kann auch vom Typ `float` sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("U.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne U-Instanz erhalten Sie mit der U-Methode `info`, beispielsweise für `u1`:

```
>>> u1.info()
Name           : U(10:11:12)
Art            : U-Instanz (Uhrzeit)
Wert (Stunden) : 10
Wert (Minuten) : 11
Wert (Sekunden): 12
interner Wert  : 36672 (in s)
-----
```

Informationen über alle U-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("U")
2 Element(e):

Name          : U(10:11:12)
Art           : U-Instanz (Uhrzeit)
Wert (Stunden) : 10
Wert (Minuten) : 11
Wert (Sekunden): 12
interner Wert  : 36672 (in s)
-----

Name          : U(20:21:22)
Art           : U-Instanz (Uhrzeit)
Wert (Stunden) : 20
Wert (Minuten) : 21
Wert (Sekunden): 22
interner Wert  : 73282 (in s)
-----
```

11.4 Operatoren

Vergleiche

Instanzen der Klasse U können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```
>>> u1; u2
10:11:12
T1(-73282,s)
>>> u1 <= u2
False
>>> u2 != U(23, 23, 23)
True
>>> u1 == U(3, 2, 1.5)
False
```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> u1
10:11:12
>>> (U(1, 0, 0) <= u1) and (u1 <= U(5, 59, 59))    # u1 zwischen 1:00 und 5:59
False
>>> not ((u1 == U(1, 0, 0)) or (u1 == U(6, 0, 0))) # u1 weder 1:00 noch 6:00
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse U verfügt mit Einschränkungen nur über die arithmetischen Operatoren + und -. Siehe dazu Tabelle 11.2.

Tabelle 11.2: Klasse U, arithmetische Operatoren und rechtsseitige Operanden

	U	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	–	–	–	–	–	–	–	–	(U)	–	–	–	–	–	–	(U)
/	–	–	–	–	–	–	–	–	(U)	–	–	–	–	–	–	(U)
+	(U)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
-	(U)	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Beispiele – arithmetische Operatoren

```
>>> u1, u2
(10:11:12, T1(-73282,s))
>>> u1 + u2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 7661, in __add__
    raise TypeError(_fehler3 + str(u))
TypeError: Operand hat keinen zulässigen Typ: T1(-73282,s)
>>> u2 - U(23, 0, 0)
T1(-156082,s)
>>> U(10, 20, 30) + U(11, 21, 31)
21:42:01
>>> u1 / u2
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for /: 'U' and 'T1'
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit U-Instanzen gibt es durch die U-Methode `U.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok(Ü.beispiel)`, beispielsweise `U.beispiel()`.

11.5 Eigenschaften der U-Instanzen

Jede U-Instanz (jedes U-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der U-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `U.classInfo("V")`. Danach stehen in der Klasse U die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> U.classInfo("V")
Eigenschaften der U-Instanzen:

u.v      Wert der Instanz   (gemessen in s)
u.u      Maßeinheit der Instanz: s
u.v1     Wert der Stunden  (gemessen in h)
u.v2     Wert der Minuten  (gemessen in minute)
u.v3     Wert der Sekunden (gemessen in s)
u.name   Name der Instanz
u.internal interner Wert
```

Beispiele

```
>>> u1.info()
Name      : U(10:11:12)
Art       : U-Instanz (Uhrzeit)
Wert (Stunden) : 10
Wert (Minuten) : 11
Wert (Sekunden): 12
interner Wert  : 36672 (in s)
-----

>>> u1.u
's'
>>> u1.v
36672
>>> u1.v1
```

```

10
>>> u1.v2
11
>>> u1.v3
12
>>> u1.name
'U(10:11:12)'
>>> u1.internal
36672

```

11.6 Konvertierung

Instanzen der Klasse U können mit der Hilfe der U-Methode `to(maßeinheit)` in eine andere Maßeinheit umgerechnet werden. Ein Aufruf der U-Methode `toYMD` liefert Jahr, Monat, Tag, Stunde, Minute, Sekunde der aktuellen U-Instanz als Liste.

Beispiele – Konvertierung

```

>>> u1
10:11:12
>>> u1.to(h)           # uhrzeit --> h
T1(10.19,h)
>>> u1.to(minute)      # uhrzeit --> minute
T1(611.2,minute)
>>> u1.to(s)           # Uhrzeit --> s
T1(36672.0,s)
>>> u1.toYMD()         # Uhrzeit --> YMD (year, month, day, ...)
[T1(0,y), T1(0,mon), T1(0,d), T1(10,h), T1(11,minute), T1(12,s)]
>>> u1.to(s2)          # s --> s2 ?
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 7538, in to
    raise ValueError(_fehler2 + str(unit))
ValueError: Unit ist hier nicht bekannt/zulässig: s2

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

12 Abgeleitete Klasse V

V ist eine Subklasse von G und erbt daher auch alle Eigenschaften (Abschnitt 9.5 auf Seite 89) und Methoden (Abschnitt 9.2 auf Seite 82) von G (Abschnitt 9 auf Seite 81). Sie realisiert das Rechnen mit mechanischen Geschwindigkeiten:

$$V = \frac{\text{Länge}}{\text{Zeit}} \left[\frac{\text{m}}{\text{s}} \right]$$

12.1 Übersicht

Globale Informationen über die Klasse V

Eine Übersicht über die Klasse V erhalten Sie durch die globale Methode `dok("V")` oder – etwas ausführlicher – mit der V-Methode `V.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("V.classInfo")`:

```
>>> dok("V.classInfo")
V.classInfo
    Gibt Informationen zur Klasse V und ihren Methoden aus.

    Aufruf: V.classInfo(m=art) oder V.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> V.classInfo("H")
Class V

    V ist eine Subklasse von G und realisiert das Rechnen mit V-Instanzen
    (Geschwindigkeiten). Sie erbt damit alle Eigenschaften und Methoden von G
    (außer denen, die überschrieben werden).

    Aufruf: V.__init__(l, t, n="")

    Parameter:

    l      : L1-Instanz;           keine Voreinstellung
```

12 Abgeleitete Klasse V

t : T1-Instanz; keine Voreinstellung
n="" : Name der V-Instanz; Voreinstellung: leere Zeichenkette

Methoden:

v : V-Instanz (Geschwindigkeit)
e : eine Methode der Klasse V
o : eine zulässige Instanz

v.__abs__() oder abs(v)	Liefert den Absolut-Betrag der V-Instanz.
v.__add__(o) oder v + o	Realisiert das Addieren von V-Instanzen.
V.__init__(l, t, n='')	Initialisiert eine V-Instanz.
v.__mul__(o) oder v * o	Realisiert das Multiplizieren von V-Instanzen.
v.__repr__() oder repr(v)	Repräsentiert eine V-Instanz.
v.__str__() oder str(v)	Repräsentiert eine V-Instanz.
v.__sub__(o) oder v - o	Realisiert das Subtrahieren von V-Instanzen.
v.__truediv__(o) oder v / o	Realisiert die Division von V-Instanzen.
v.__eq__(v) oder v == v	Realisiert den Vergleich == in V.
v.__ge__(v) oder v >= v	Realisiert den Vergleich >= in V.
v.__gt__(v) oder v > v	Realisiert den Vergleich > in V.
v.__le__(v) oder v <= v	Realisiert den Vergleich <= in V.
v.__lt__(v) oder v < v	Realisiert den Vergleich < in V.
v.__ne__(v) oder v != v	Realisiert den Vergleich != in V.
V.classInfo(m=art)	Gibt Informationen zur Klasse V und ihren Methoden aus.
V.ci(m=art)	Alias für V.classInfo(m=art)
V.beispiel(e)	Liefert Beispiele für die Methoden der Klasse V.
V.description()	Liefert eine Kurzbeschreibung der Klasse V.
v.to(liste)	Gibt in geeigneter Weise den Wert einer V-Instanz aus.
v.info()	Gibt Informationen über eine V-Instanz (Geschwindigkeit) aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt [12.2 auf der nächsten Seite](#)
- Maßeinheiten und Instanzen → Abschnitt [12.3 auf Seite 108](#)
- Eigenschaften der Instanzen → Abschnitt [12.5 auf Seite 112](#)
- Zulässige Operationen → Abschnitt [12.4 auf Seite 110](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [12.6 auf Seite 113](#)

12.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("V")` auf Seite 105 liefert u. a. eine Auflistung der V-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 12.1 aufgeführt.

Tabelle 12.1: Methoden der Klasse V

Name	Bedeutung
<code>V.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse V.
<code>V.classInfo(art)</code>	Gibt Informationen zur Klasse V und ihren Methoden aus.
<code>V.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>V.description()</code>	Liefert eine einzeilige Kurzbeschreibung der Klasse V.
<code>v.info(modus)</code>	Gibt Informationen über die V-Instanz <i>v</i> (Geschwindigkeit) aus.

Ausführlichere Informationen über eine bestimmte V-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `V.classInfo("M")` für alle V-Methoden.

Beispiele für dok

```
>>> dok("V.to")
V.to
    Gibt in geeigneter Weise den Wert einer G-Instanz aus.

    Aufruf: g.to(wliste)
    vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
    - leer : keine Konvertierung [Voreinstellung], aber Bereinigung
    - kmh  : m --> km, s --> h
    - mks  : m --> m, s --> s, kg --> kg
    - cgs  : m --> cm, kg --> g, s --> s
    - mph  : m --> mi, s --> h
    mögliche Fehlermeldung:
    - Parameter ist unzulässig.

>>> dok("V.info")
V.info
    Gibt Informationen über eine V-Instanz (Geschwindigkeit) aus.

    Aufruf: v.info()
```

12.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse V besitzt keine eigenen Maßeinheiten; sie nutzt die von L1 (Abschnitt 2.3 auf Seite 18) und T1 (Abschnitt 5.3 auf Seite 48).

Instanzen

Eigene V-Instanzen können Sie mittels der Anweisung `V(länge,zeit,n=name)` oder der globalen Methode `meterS(wert)` (beide Geschwindigkeit) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> l11
L1(2,m)
>>> t11
T1(2,s)
>>> v1 = V(l11, t11)
>>> v2 = +meterS(3.75)                # globale Methode
>>> v3 = V(l11*2, t11, n="doppelt")
>>> v4 = -V(l11/2, t11*2)
>>> v1
V(L1(2,m), T1(2,s))
>>> v2
V(L1(3.75,m), T1(1,s))
>>> v3
V(L1(4,m), T1(2,s))
>>> v4
G(L1(-1.0,m) / (T1(4,s)))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene V-Instanz zu generieren, kann misslingen:

- ☐ **Operand hat den Wert Null:** Der zweite Parameter *zeit* darf nicht den Wert Null haben.
- ☐ **Operand hat keinen zulässigen Typ:** Der erste Parameter *länge* muss vom Typ L1, der zweite *zeit* vom Typ T1 sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("V.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne V-Instanz erhalten Sie mit Hilfe der V-Methode `info`, beispielsweise für `v2`:

```
>>> v2.info()
Name           : meterS(3.75)
Art            : V-Instanz (Geschwindigkeit)
linker Operand : L1(3.75,m)
```

12 Abgeleitete Klasse V

```
Operator          : /
rechter Operand   : T1(1,s)
Wert              : 3.75 (in den angegebenen Maßeinheiten)
interner Wert     : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 13.5 ['km']/['h']
konvertiert (cgs)  : 375.0 ['cm']/['s']
konvertiert (mks)  : 3.75 ['m']/['s']
-----
```

Informationen über alle V-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("V")
5 Element(e):
```

```
Name          : meterS(3)
Art           : V-Instanz (Geschwindigkeit)
linker Operand : L1(3,m)
Operator      : /
rechter Operand : T1(1,s)
Wert          : 3.0 (in den angegebenen Maßeinheiten)
interner Wert  : 3.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 10.8 ['km']/['h']
konvertiert (cgs)  : 300.0 ['cm']/['s']
konvertiert (mks)  : 3.0 ['m']/['s']
-----
```

```
Name          : V(L1(2,m), T1(2,s))
Art           : V-Instanz (Geschwindigkeit)
linker Operand : L1(2,m)
Operator      : /
rechter Operand : T1(2,s)
Wert          : 1.0 (in den angegebenen Maßeinheiten)
interner Wert  : 1.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 3.6 ['km']/['h']
konvertiert (cgs)  : 100.0 ['cm']/['s']
konvertiert (mks)  : 1.0 ['m']/['s']
-----
```

```
Name          : meterS(3.75)
Art           : V-Instanz (Geschwindigkeit)
linker Operand : L1(3.75,m)
```

12 Abgeleitete Klasse V

```
Operator          : /
rechter Operand   : T1(1,s)
Wert              : 3.75 (in den angegebenen Maßeinheiten)
interner Wert     : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 13.5 ['km']/['h']
konvertiert (cgs)  : 375.0 ['cm']/['s']
konvertiert (mks)  : 3.75 ['m']/['s']
-----
```

```
Name              : doppelt
Art                : V-Instanz (Geschwindigkeit)
linker Operand     : L1(4,m)
Operator           : /
rechter Operand    : T1(2,s)
Wert               : 2.0 (in den angegebenen Maßeinheiten)
interner Wert      : 2.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 7.2 ['km']/['h']
konvertiert (cgs)  : 200.0 ['cm']/['s']
konvertiert (mks)  : 2.0 ['m']/['s']
-----
```

```
Name              : V(L1(1.0,m), T1(4,s))
Art                : V-Instanz (Geschwindigkeit)
linker Operand     : L1(1.0,m)
Operator           : /
rechter Operand    : T1(4,s)
Wert               : 0.25 (in den angegebenen Maßeinheiten)
interner Wert      : 0.25 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h) : 0.9 ['km']/['h']
konvertiert (cgs)  : 25.0 ['cm']/['s']
konvertiert (mks)  : 0.25 ['m']/['s']
-----
```

12.4 Operatoren

Vergleiche

Instanzen der Klasse V können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«

2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```
>>> v1; v2; v3
V(L1(2,m), T1(2,s))
V(L1(3.75,m), T1(1,s))
V(L1(4,m), T1(2,s))
>>> v1 <= v2
True
>>> v3 >= v1
True
>>> v2 != meterS(23) # globale Methode
True
>>> v3 == meterS(3.4) # globale Methode
False
```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> v1
V(L1(2,m), T1(2,s))
>>> (meterS(1) <= v1) and (v1 <= meterS(5)) # v1 zwischen 1m/s und 5m/s
True
>>> not ((v1 == meterS(1)) or (v1 == meterS(5))) # v1 weder 1m/s noch 5m/s
False
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse V verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle [12.2 auf der nächsten Seite](#).

Beispiele – arithmetische Operatoren

```
>>> v1; v2; v3
V(L1(2,m), T1(2,s))
V(L1(3.75,m), T1(1,s))
V(L1(4,m), T1(2,s))
>>> v1 + v2
```

Tabelle 12.2: Klasse V, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	G	G	L1	G	G	G	G	V
/	T1	G	G	G	G	G	G	V	G	B	G	G	Skalar	G	V
+	—	—	—	—	—	—	—	V	—	—	—	—	—	—	—
-	—	—	—	—	—	—	—	V	—	—	—	—	—	—	—
**	—	—	—	—	—	—	—	?	—	—	—	—	—	—	?

```

V(L1(4.75,m), T1(1,s))
>>> v3 - v2
V(L1(-1.75,m), T1(1,s))
>>> v2 * T1(3.4, h)
L1(45900.0,m)
>>> v1 / v2
0.26666666666666666
>>> v3 ** 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'V' and 'int'

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit V-Instanzen erhalten Sie durch die Methode `V.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("V.beispiel")`, beispielsweise `V.beispiel()`.

12.5 Eigenschaften der V-Instanzen

Jede V-Instanz (jedes V-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der V-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `V.classInfo("V")`. Danach stehen in der Klasse V die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> V.classInfo("V")
Eigenschaften der V-Instanzen:

v.links      linker Operand
v.operator   Operator
v.rechts     rechter Operand
v.name       Name der Instanz
v.oben       Einheiten im Zähler

```

```

v.unten    Einheiten im Nenner
v.internal interner Wert
v.v        Wert in den angegebenen Einheiten

```

Beispiele

```

>>> v3
V(L1(4,m), T1(2,s))
>>> v3.info()
Name           : doppelt
Art            : V-Instanz (Geschwindigkeit)
linker Operand : L1(4,m)
Operator       : /
rechter Operand: T1(2,s)
Wert           : 2.0 (in den angegebenen Maßeinheiten)
interner Wert  : 2.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s']
konvertiert (km/h)  : 7.2 ['km']/['h']
konvertiert (cgs)   : 200.0 ['cm']/['s']
konvertiert (mks)   : 2.0 ['m']/['s']
-----

>>> v3.links
L1(4,m)
>>> v3.operator
 '/'
>>> v3.rechts
T1(2,s)
>>> v3.name
'doppelt'
>>> v3.oben
['m']
>>> v3.unten
['s']
>>> v3.internal
2.0
>>> v3.v
2.0

```

12.6 Konvertierung

Instanzen der Klasse `V` können mit der Hilfe der `V`-Methode `to(einheitensystem)` in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

$leer$: keine Konvertierung, aber Bereinigung
 $kmh : m \rightarrow km, s \rightarrow h$
 $mks : m \rightarrow m, s \rightarrow s, kg \rightarrow kg$
 $cgs : m \rightarrow cm, kg \rightarrow g, s \rightarrow s$
 $mph : m \rightarrow mi, s \rightarrow h$

Beispiele – Konvertierung

```

>>> v1; v2; v3
V(L1(2,m), T1(2,s))
V(L1(3.75,m), T1(1,s))
V(L1(4,m), T1(2,s))
>>> v1.to(cgs)
"100.0 ['cm']/['s']"
>>> v1.to(mks)
"1.0 ['m']/['s']"
>>> v1.to(kmh)
"3.6 ['km']/['h']"
>>> v1.to(mph)
"2.24 ['mi']/['h']"
>>> v2.to(kmh)
"13.5 ['km']/['h']"
>>> V(L1(200,m),T1(19.9,s)).to(kmh)
"36.18 ['km']/['h']"
>>> v3.to(cgs)
"200.0 ['cm']/['s']"
>>> meterS(330).to(kmh)           # globale Methode
"1188.0 ['km']/['h']"
>>> v1.to(hkm)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 2144, in to
    raise ValueError(_fehler5 + str(d))
ValueError: Parameter ist unzulässig: hkm

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

13 Abgeleitete Klasse B

B ist eine Subklasse von G und erbt daher auch alle Eigenschaften (Abschnitt 9.5 auf Seite 89) und Methoden (Abschnitt 9.2 auf Seite 82) von G (Abschnitt 9 auf Seite 81). Sie realisiert das Rechnen mit Beschleunigungen:

$$W = \frac{\text{Länge}}{\text{Zeit}^2} \left[\frac{\text{m}}{\text{s}^2} \right]$$

13.1 Übersicht

Globale Informationen über die Klasse B

Eine Übersicht über die Klasse B erhalten Sie durch die globale Methode `dok("B")` oder – etwas ausführlicher – mit der B-Methode `B.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("B.classInfo")`:

```
>>> dok("B.classInfo")
B.classInfo
    Gibt Informationen zur Klasse B und ihren Methoden aus.

    Aufruf: B.classInfo(m=art) oder B.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> B.classInfo("H")
Class B

    B ist eine Subklasse von G und realisiert das Rechnen mit B-Instanzen
    (Beschleunigung). Sie erbt damit alle Eigenschaften und Methoden von G
    (außer denen, die überschrieben werden).

    Aufruf: B.__init__(l, t2, n="")

    Parameter:

    l      : L1-Instanz;           keine Voreinstellung
    t2     : T2-Instanz;           keine Voreinstellung
```

13 Abgeleitete Klasse B

`n=""` : Name der B-Instanz; Voreinstellung: leere Zeichenkette

Methoden:

`b` : B-Instanz (Beschleunigung)

`e` : eine Methode der Klasse B

`o` : eine zulässige Instanz

<code>b.__abs__()</code> oder <code>abs(b)</code>	Liefert den Absolut-Betrag der B-Instanz.
<code>b.__add__(o)</code> oder <code>b + o</code>	Realisiert das Addieren von B-Instanzen.
<code>B.__init__(l, t2, n='')</code>	Initialisiert eine B-Instanz.
<code>b.__mul__(o)</code> oder <code>b * o</code>	Realisiert das Multiplizieren von B-Instanzen.
<code>b.__repr__()</code> oder <code>repr(b)</code>	Repräsentiert eine B-Instanz.
<code>b.__str__()</code> oder <code>str(b)</code>	Repräsentiert eine B-Instanz.
<code>b.__sub__(o)</code> oder <code>b - o</code>	Realisiert das Subtrahieren von B-Instanzen.
<code>b.__truediv__(o)</code> oder <code>b / o</code>	Realisiert die Division von B-Instanzen.
<code>b.__eq__(b)</code> oder <code>b == b</code>	Realisiert den Vergleich <code>==</code> in B.
<code>b.__ge__(b)</code> oder <code>b >= b</code>	Realisiert den Vergleich <code>>=</code> in B.
<code>b.__gt__(b)</code> oder <code>b > b</code>	Realisiert den Vergleich <code>></code> in B.
<code>b.__le__(b)</code> oder <code>b <= b</code>	Realisiert den Vergleich <code><=</code> in B.
<code>b.__lt__(b)</code> oder <code>b < b</code>	Realisiert den Vergleich <code><</code> in B.
<code>b.__ne__(b)</code> oder <code>b != b</code>	Realisiert den Vergleich <code>!=</code> in B.
<code>B.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse B.
<code>B.description()</code>	Liefert eine Kurzbeschreibung von B.
<code>B.ci(m=art)</code>	Alias für <code>B.classInfo(m=art)</code>
<code>B.classInfo(m=art)</code>	Gibt Informationen zur Klasse B und ihren Methoden aus.
<code>b.to(liste)</code>	Gibt in geeigneter Weise den Wert einer B-Instanz aus.
<code>b.info()</code>	Gibt Informationen über eine B-Instanz (Beschleunigung) aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt [13.2](#)
- Maßeinheiten und Instanzen → Abschnitt [13.3 auf der nächsten Seite](#)
- Eigenschaften der Instanzen → Abschnitt [13.5 auf Seite 122](#)
- Zulässige Operationen → Abschnitt [13.4 auf Seite 120](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [13.6 auf Seite 123](#)

13.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("B")` auf der vorherigen Seite liefert u. a. eine Auflistung der B-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der

Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 13.1 aufgeführt.

Tabelle 13.1: Methoden der Klasse B

Name	Bedeutung
<code>B.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse B.
<code>B.classInfo(art)</code>	Gibt Informationen zur Klasse B und ihren Methoden aus.
<code>B.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>B.description()</code>	Liefert eine einzeilige Kurzbeschreibung von B.
<code>b.info(modus)</code>	Gibt Informationen über die B-Instanz <i>b</i> (Beschleunigung) aus.

Ausführlichere Informationen über eine bestimmte B-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `B.classInfo("M")` für alle B-Methoden.

Beispiele für dok

```
>>> dok("B.to")
B.to
    Gibt in geeigneter Weise den Wert einer G-Instanz aus.

    Aufruf: g.to(wliste)
    vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
    - leer : keine Konvertierung [Voreinstellung], aber Bereinigung
    - kmh  : m --> km, s --> h
    - mks  : m --> m, s --> s, kg --> kg
    - cgs  : m --> cm, kg --> g, s --> s
    - mph  : m --> mi, s --> h
    mögliche Fehlermeldung:
    - Parameter ist unzulässig.

>>> dok("B.info")
B.info
    Gibt Informationen über eine B-Instanz (Beschleunigung) aus.

    Aufruf: b.info()
```

13.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse B besitzt keine eigenen Maßeinheiten; sie nutzt die von L1 (Abschnitt 2.3 auf Seite 18) und T2 (Abschnitt 6.3 auf Seite 57).

Instanzen

Eigene B-Instanzen können Sie mittels der Anweisung `B(länge, zeit2, n=name)` oder der globalen Methode `meterS2(wert)` (Beschleunigung) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> l12
L1(-2,mm)
>>> t21
T2(2,s2)
>>> b1 = B(l12, t21)
>>> b2 = meterS2(3.75)           # globale Methode
>>> b3 = +B(l12, t21 * 3, n="drittel")
>>> b4 = -B(l12/2, t21*2)
>>> b1
B(L1(-2,mm), T2(2,s2))
>>> b2
B(L1(3.75,m), T2(1,s2))
>>> b3
B(L1(-2,mm), T2(6,s2))
>>> b4
G(L1(0.0,mm) / (T2(4,s2)))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene B-Instanz zu generieren, kann misslingen:

- ☐ **Operand hat den Wert Null:** Der zweite Parameter *zeit*² darf nicht den Wert Null haben.
- ☐ **Operand hat keinen zulässigen Typ:** Der erste Parameter *länge* muss vom Typ L1, der zweite *zeit*² vom Typ T2 sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok{"B.__init__"}`.

Informationen über Instanzen

Informationen über eine einzelne B-Instanz erhalten Sie mit Hilfe der B-Methode `info`, beispielsweise für `b1`:

```
>>> b1.info()
Name           : B(L1(-2,mm), t)
Art            : B-Instanz (Beschleunigung)
linker Operand : L1(-2,mm)
Operator       : /
rechter Operand : T2(2,s2)
Wert           : -1.0 (in den angegebenen Maßeinheiten)
interner Wert  : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h) : -12.96 ['km']/['h', 'h']
```

13 Abgeleitete Klasse B

```
konvertiert (cgs)      : -0.1 ['cm']/['s', 's']
konvertiert (mks)      : -0.0 ['m']/['s', 's']
-----
```

Informationen über alle B-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("B")
4 Element(e):
```

```
Name          : B(L1(-2,mm), t)
Art            : B-Instanz (Beschleunigung)
linker Operand : L1(-2,mm)
Operator       : /
rechter Operand: T2(2,s2)
Wert           : -1.0 (in den angegebenen Maßeinheiten)
interner Wert  : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h) : -12.96 ['km']/['h', 'h']
konvertiert (cgs)  : -0.1 ['cm']/['s', 's']
konvertiert (mks)  : -0.0 ['m']/['s', 's']
-----
```

```
Name          : meterS2(3.75)
Art            : B-Instanz (Beschleunigung)
linker Operand : L1(3.75,m)
Operator       : /
rechter Operand: T2(1,s2)
Wert           : 3.75 (in den angegebenen Maßeinheiten)
interner Wert  : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['m']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h) : 48600.0 ['km']/['h', 'h']
konvertiert (cgs)  : 375.0 ['cm']/['s', 's']
konvertiert (mks)  : 3.75 ['m']/['s', 's']
-----
```

```
Name          : drittel
Art            : B-Instanz (Beschleunigung)
linker Operand : L1(-2,mm)
Operator       : /
rechter Operand: T2(6,s2)
Wert           : -0.33 (in den angegebenen Maßeinheiten)
interner Wert  : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h) : -4.32 ['km']/['h', 'h']
```

```

konvertiert (cgs)      : -0.03 ['cm']/['s', 's']
konvertiert (mks)      : -0.0 ['m']/['s', 's']
-----

Name                   : B(L1(-0.0,mm), t)
Art                    : B-Instanz (Beschleunigung)
linker Operand         : L1(-0.0,mm)
Operator               : /
rechter Operand        : T2(4,s2)
Wert                   : -0.0 (in den angegebenen Maßeinheiten)
interner Wert          : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h)     : -0.0 ['km']/['h', 'h']
konvertiert (cgs)      : -0.0 ['cm']/['s', 's']
konvertiert (mks)      : -0.0 ['m']/['s', 's']
-----

```

13.4 Operatoren

Vergleiche

Instanzen der Klasse B können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> b1; b2; b3
B(L1(-2,mm), T2(2,s2))
B(L1(3.75,m), T2(1,s2))
B(L1(-2,mm), T2(6,s2))
>>> b1 <= b2
True
>>> b3 >= b1
True
>>> b1 != meterS2(23)    # globale Methode
True
>>> b3 == meterS2(3.4)   # globale Methode
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> b1
B(L1(-2,mm), T2(2,s2))
>>> (meterS2(1) <= b1) and (b1 <= meterS2(5))      # b1 zwischen 1m/s2 und 5m/s2
False
>>> not ((b1 == meterS2(1)) or (b1 == meterS2(5))) # b1 weder 1m/s2 noch 5m/s2
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse B verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 13.2.

Tabelle 13.2: Klasse B, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	F1	B	G	V	L1	G	G	G	B
/	Skalar	G	G	G	G	G	G	B	G	G	G	G	G	G	B
+	B	–	–	–	–	–	–	B	–	–	–	–	–	–	–
-	B	–	–	–	–	–	–	B	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```
>>> b1; b2; b3
B(L1(-2,mm), T2(2,s2))
B(L1(3.75,m), T2(1,s2))
B(L1(-2,mm), T2(6,s2))
>>> b1 + b2
B(L1(3.75,m), T2(1,s2))
>>> b3 - meterS2(23) # globale Methode
B(L1(-23.0,m), T2(1,s2))
>>> b1 * T(3.4, h)
V(L1(-12.24,m), T1(1,s))
>>> b1 / b2
-0.00026666666666666667
>>> b3 ** 2
Traceback (most recent call last):
```

```
File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'B' and 'int'
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit B-Instanzen erhalten Sie durch die Methode `B.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("B.beispiel")`, beispielsweise `B.beispiel()`.

13.5 Eigenschaften der B-Instanzen

Jede B-Instanz (jedes B-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der B-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `B.classInfo("V")`. Danach stehen in der Klasse B die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> B.classInfo("V")
Eigenschaften der B-Instanzen:

b.links      linker Operand
b.operator   Operator
b.rechts     rechter Operand
b.name       Name der Instanz
b.oben       Einheiten im Zähler
b.unten      Einheiten im Nenner
b.internal   interner Wert
b.v          Wert in den angegebenen Einheiten
```

Beispiele

```
>>> b3
B(L1(-2,mm), T2(6,s2))
>>> b3.info()
Name           : drittel
Art            : B-Instanz (Beschleunigung)
linker Operand : L1(-2,mm)
Operator       : /
rechter Operand : T2(6,s2)
Wert           : -0.33 (in den angegebenen Maßeinheiten)
interner Wert  : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (km/h) : -4.32 ['km']/['h', 'h']
```



```

konvertiert (cgs)      : -0.03 ['cm']/['s', 's']
konvertiert (mks)      : -0.0 ['m']/['s', 's']
-----

```

```

>>> b3.links
L1(-2,mm)
>>> b3.operator
 '/'
>>> b3.rechts
T2(6,s2)
>>> b3.name
'drittel'
>>> b3.oben
['mm']
>>> b3.unten
['s2']
>>> b3.internal
-0.00033333333333333333
>>> b3.v
-0.3333333333333333

```

13.6 Konvertierung

Instanzen der Klasse B können mit der Hilfe der B-Methode `to(einheitensystem)` in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

leer : keine Konvertierung, aber Vereinigung
 $\text{kmh} : \text{m} \longrightarrow \text{km}, \text{s} \longrightarrow \text{h}$
 $\text{mks} : \text{m} \longrightarrow \text{m}, \text{s} \longrightarrow \text{s}, \text{kg} \longrightarrow \text{kg}$
 $\text{cgs} : \text{m} \longrightarrow \text{cm}, \text{kg} \longrightarrow \text{g}, \text{s} \longrightarrow \text{s}$
 $\text{mph} : \text{m} \longrightarrow \text{mi}, \text{s} \longrightarrow \text{h}$

Beispiele – Konvertierung

```

>>> b1; b2; b3
B(L1(-2,mm), T2(2,s2))
B(L1(3.75,m), T2(1,s2))
B(L1(-2,mm), T2(6,s2))
>>> b1.to(cgs)
"-0.1 ['cm']/['s', 's']"
>>> b1.to(mks)
"-0.0 ['m']/['s', 's']"
>>> b1.to(kmh)
"-12.96 ['km']/['h', 'h']"
>>> b1.to(mph)
"-8.05 ['mi']/['h', 'h']"

```

```

>>> b2.to(kmh)
"48600.0 ['km']/['h', 'h']"
>>> b3.to(cgs)
"-0.03 ['cm']/['s', 's']"
>>> meterS2(2.5).to(kmh) # globale Methode
"32400.0 ['km']/['h', 'h']"
>>> meterS2(35).to(cgs) # globale Methode
"3500.0 ['cm']/['s', 's']"
>>> b1.to(hkm)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 2144, in to
    raise ValueError(_fehler5 + str(d))
ValueError: Parameter ist unzulässig: hkm

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

14 Abgeleitete Klasse F1

F1 ist eine Subklasse von G und erbt daher auch alle Eigenschaften (Abschnitt 9.5 auf Seite 89) und Methoden (Abschnitt 9.2 auf Seite 82) von G (Abschnitt 9 auf Seite 81). Sie realisiert das Rechnen mit mechanischen Kräften:

$$F_1 = \text{Masse} \times \text{Beschleunigung} \left[\text{kg} \times \frac{\text{m}}{\text{s}^2} \right]$$

14.1 Übersicht

Globale Informationen über die Klasse F1

Eine Übersicht über die Klasse F1 erhalten Sie durch die globale Methode `dok("F1")` oder – etwas ausführlicher – mit der F1-Methode `F1.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch die globale Methode `dok("F1.classInfo")`:

```
>>> dok("F1.classInfo")
F1.classInfo
    Gibt Informationen zur Klasse F1 und ihren Methoden aus.

    Aufruf: F1.classInfo(m=art) oder F1.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> F1.classInfo("H")
Class F1

    F1 ist eine Subklasse von G und realisiert das Rechnen mit F1-Instanzen
    (Kraft). Sie erbt damit alle Eigenschaften und Methoden von G (außer denen,
    die überschrieben werden).

    Aufruf: F1.__init__(m, b, n="")

    Parameter:

    m      : M-Instanz;           keine Voreinstellung
```

14 Abgeleitete Klasse F1

b : B-Instanz; keine Voreinstellung
n="" : Name der F1-Instanz; Voreinstellung: leere Zeichenkette

Methoden:

f : F1-Instanz (Kraft)
e : eine Methode der Klasse F1
o : ein zulässige Instanz

f.__abs__() oder abs(f)	Liefert den Absolut-Betrag der F1-Instanz.
f.__add__(o) oder f + o	Realisiert das Addieren von F1-Instanzen.
f.__mul__(o) oder f * o	Realisiert das Multiplizieren von F1-Instanzen.
f.__repr__() oder repr(f)	Repräsentiert eine F1-Instanz.
f.__str__() oder str(f)	Repräsentiert eine F1-Instanz.
f.__sub__(o) oder f - o	Realisiert das Subtrahieren von F1-Instanzen.
f.__truediv__(o) oder f / o	Realisiert die Division von F1-Instanzen.
F1.__init__(m, b, n='')	Initialisiert eine F1-Instanz.
f.__eq__(f) oder f == f	Realisiert den Vergleich == in F1.
f.__ge__(f) oder f >= f	Realisiert den Vergleich >= in F1.
f.__gt__(f) oder f > f	Realisiert den Vergleich > in F1.
f.__le__(f) oder f <= f	Realisiert den Vergleich <= in F1.
f.__lt__(f) oder f < f	Realisiert den Vergleich < in F1.
f.__ne__(f) oder f != f	Realisiert den Vergleich != in F1.
F1.beispiel(e)	Liefert Beispiele für die Methoden der Klasse F1.
F1.classInfo(m=art)	Gibt Informationen zur Klasse F1 und ihren Methoden aus.
F1.ci(m=art)	Alias für F1.classInfo(m=art)
f.to(liste)	Gibt in geeigneter Weise den Wert einer F1-Instanz aus.
F1.description()	Liefert eine Kurzbeschreibung der Klasse F1.
F1.info()	Gibt Informationen über eine F1-Instanz (Kraft) aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt [14.2](#)
- Maßeinheiten und Instanzen → Abschnitt [14.3 auf der nächsten Seite](#)
- Eigenschaften der Instanzen → Abschnitt [14.5 auf Seite 132](#)
- Zulässige Operationen → Abschnitt [14.4 auf Seite 130](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [14.6 auf Seite 133](#)

14.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("F1")` auf der vorherigen Seite liefert u. a. eine Auflistung der F1-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der

Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 14.1 aufgeführt.

Tabelle 14.1: Methoden der Klasse F1

Name	Bedeutung
<code>F1.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse F1.
<code>F1.classInfo(art)</code>	Gibt Informationen zur Klasse F1 und ihren Methoden aus.
<code>F1.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>F1.description()</code>	Liefert eine einzeilige Kurzbeschreibung der Klasse F1.
<code>f.info(modus)</code>	Gibt Informationen über die F1-Instanz <i>f</i> (Kraft) aus.

Ausführlichere Informationen über eine bestimmte F1-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `F1.classInfo("M")` für alle F1-Methoden.

Beispiele für dok

```
>>> dok("F1.to")
F1.to
    Gibt in geeigneter Weise den Wert einer G-Instanz aus.

    Aufruf: g.to(wliste)
    vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
    - leer : keine Konvertierung [Voreinstellung], aber Bereinigung
    - kmh  : m --> km, s --> h
    - mks  : m --> m, s --> s, kg --> kg
    - cgs  : m --> cm, kg --> g, s --> s
    - mph  : m --> mi, s --> h
    mögliche Fehlermeldung:
    - Parameter ist unzulässig.

>>> dok("F1.info")
F1.info
    Gibt Informationen über eine F1-Instanz (Kraft) aus.

    Aufruf: F1.info()
```

14.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse F1 besitzt keine eigenen Maßeinheiten; sie nutzt die von M (Abschnitt 7.3 auf Seite 66), L1 (Abschnitt 2.3 auf Seite 18) und T2 (Abschnitt 6.3 auf Seite 57).

Instanzen

Eigene F1-Instanzen können Sie mittels der Anweisung `F1(masse, beschleunigung, n=name)` oder der globalen Methode `newton(wert)` (beide Kraft) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> m11
M(2,kg)
>>> b1
B(L1(-2,mm), T2(2,s2))
>>> f1 = F1(m11, b1)
>>> f2 = F1(m11, b1 * 3, n="dreifach")
>>> f3 = -newton(3.75) # globale Methode
>>> f4 = +newton(3.75 * 2, n="doppelt") # globale Methode
>>> f1
F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
>>> f2
F1(M(2,kg), B(L1(-6,mm), T2(2,s2)))
>>> f3
G(M(-3.75,kg) * (B(L1(1,m), T2(1,s2))))
>>> f4
F1(M(7.5,kg), B(L1(1,m), T2(1,s2)))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene F1-Instanz zu generieren, kann misslingen:

- **Operand hat keinen zulässigen Typ:** Der erste Parameter *masse* muss vom Typ M, der zweite *beschleunigung* vom Typ B sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("F1.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne F1-Instanz erhalten Sie mit Hilfe der F1-Methode `info`, beispielsweise für `f2`

```
>>> f2.info()
Name           : dreifach
Art            : F1-Instanz (Kraft)
linker Operand : M(2,kg)
Operator       : *
rechter Operand: B(L1(-6,mm), T2(2,s2))
Wert           : -6.0 (in den angegebenen Maßeinheiten)
interner Wert  : -0.01 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs) : -600.0 ['g', 'cm']/['s', 's']
```

14 Abgeleitete Klasse F1

```
konvertiert (mks) [N]: -0.01 ['kg', 'm']/['s', 's']
```

Informationen über alle F1-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("F1")
```

```
4 Element(e):
```

```
Name          : F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
Art            : F1-Instanz (Kraft)
linker Operand : M(2,kg)
Operator       : *
rechter Operand: B(L1(-2,mm), T2(2,s2))
Wert           : -2.0 (in den angegebenen Maßeinheiten)
interner Wert  : -0.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs) : -200.0 ['g', 'cm']/['s', 's']
konvertiert (mks) [N]: -0.0 ['kg', 'm']/['s', 's']
```

```
Name          : dreifach
Art            : F1-Instanz (Kraft)
linker Operand : M(2,kg)
Operator       : *
rechter Operand: B(L1(-6,mm), T2(2,s2))
Wert           : -6.0 (in den angegebenen Maßeinheiten)
interner Wert  : -0.01 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'mm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs) : -600.0 ['g', 'cm']/['s', 's']
konvertiert (mks) [N]: -0.01 ['kg', 'm']/['s', 's']
```

```
Name          : newton(3.75)
Art            : F1-Instanz (Kraft)
linker Operand : M(3.75,kg)
Operator       : *
rechter Operand: B(L1(1,m), T2(1,s2))
Wert           : 3.75 (in den angegebenen Maßeinheiten)
interner Wert  : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs) : 375000.0 ['g', 'cm']/['s', 's']
konvertiert (mks) [N]: 3.75 ['kg', 'm']/['s', 's']
```

```

Name           : doppelt
Art            : F1-Instanz (Kraft)
linker Operand : M(7.5,kg)
Operator       : *
rechter Operand: B(L1(1,m), T2(1,s2))
Wert           : 7.5 (in den angegebenen Maßeinheiten)
interner Wert  : 7.5 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs) : 750000.0 ['g', 'cm']/['s', 's']
konvertiert (mks) [N]: 7.5 ['kg', 'm']/['s', 's']
-----

```

14.4 Operatoren

Vergleiche

Instanzen der Klasse F1 können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```

>>> f1; f2; f3; f4
F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
F1(M(2,kg), B(L1(-6,mm), T2(2,s2)))
G(M(-3.75,kg) * (B(L1(1,m), T2(1,s2))))
F1(M(7.5,kg), B(L1(1,m), T2(1,s2)))
>>> f1 <= f2
False
>>> f2 >= f4
False
>>> f2 != newton(23) # globale Methode
True
>>> f4 == newton(3.4) # globale Methode
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> f1
F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
>>> (newton(1) <= f1) and (f1 <= newton(5))      # f1 zwischen 1N und 5N
False
>>> not ((f1 == newton(1)) or (f1 == newton(5))) # f1 weder 1N noch 5N
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse F1 verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 14.2.

Tabelle 14.2: Klasse F1, arithmetischen Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	W	G	G	G	F1	G	G	G	G	P	G	F1
/	M	Skalar	G	G	G	G	B	F1	G	G	G	G	G	G	F1
+	–	F1	–	–	–	–	–	F1	–	–	–	–	–	–	–
-	–	F!	–	–	–	–	–	F1	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```
>>> f1; f2; f3
F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
F1(M(2,kg), B(L1(-6,mm), T2(2,s2)))
G(M(-3.75,kg) * (B(L1(1,m), T2(1,s2))))
>>> f1 + f2
F1(M(-0.01,kg), B(L1(1,m), T2(1,s2)))
>>> f2 * T1(23, s)
G(F1(M(2,kg), B(L1(-6,mm), T2(2,s2))) * (T1(23,s)))
>>> f1 * T2(3.4, h2)
G(F1(M(2,kg), B(L1(-2,mm), T2(2,s2))) * (T2(3.4,h2)))
>>> f1 / f2
333333.3333333333
>>> f1 ** 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'F1' and 'int'
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit F1-Instanzen gibt es durch die Methode `F1.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("F1.beispiel")`, beispielsweise `F1.beispiel()`.

14.5 Eigenschaften der F1-Instanzen

Jede F1-Instanz (jedes F1-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der F1-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `F1.classInfo("V")`. Danach stehen in der Klasse F1 die folgenden Eigenschaften für Instanzen zur Verfügung:

```
>>> F1.classInfo("V")
```

Eigenschaften der F1-Instanzen:

```
f.links      linker Operand
f.operator    Operator
f.rechts      rechter Operand
f.name        Name der Instanz
f.oben        Einheiten im Zähler
f.unten       Einheiten im Nenner
f.internal    interner Wert
f.v           Wert in den angegebenen Einheiten
```

Beispiele

```
>>> f4
```

```
F1(M(7.5,kg), B(L1(1,m), T2(1,s2)))
```

```
>>> f4.info()
```

```
Name                : doppelt
Art                  : F1-Instanz (Kraft)
linker Operand       : M(7.5,kg)
Operator             : *
rechter Operand      : B(L1(1,m), T2(1,s2))
Wert                 : 7.5 (in den angegebenen Maßeinheiten)
interner Wert        : 7.5 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler: ['kg', 'm']
Maßeinheit(en) Nenner: ['s2']
konvertiert (cgs)    : 750000.0 ['g', 'cm']/['s', 's']
konvertiert (mks) [N]: 7.5 ['kg', 'm']/['s', 's']
-----
```

```
>>> f4.links
```

```
M(7.5,kg)
```

```

>>> f4.operator
'*'
>>> f4.rechts
B(L1(1,m), T2(1,s2))
>>> f4.name
'doppelt'
>>> f4.oben
['kg', 'm']
>>> f4.unten
['s2']
>>> f4.internal
7.5
>>> f4.v
7.5

```

14.6 Konvertierung

Instanzen der Klasse F1 können mit der Hilfe der F1-Methode `to(einheitensystem)` in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

leer : keine Konvertierung, aber Bereinigung

$\text{kmh} : \text{m} \rightarrow \text{km}, \text{s} \rightarrow \text{h}$

$\text{mks} : \text{m} \rightarrow \text{m}, \text{s} \rightarrow \text{s}, \text{kg} \rightarrow \text{kg}$

$\text{cgs} : \text{m} \rightarrow \text{cm}, \text{kg} \rightarrow \text{g}, \text{s} \rightarrow \text{s}$

$\text{mph} : \text{m} \rightarrow \text{mi}, \text{s} \rightarrow \text{h}$

Beispiele – Konvertierung

```

>>> f1; f2; f3; f4
F1(M(2,kg), B(L1(-2,mm), T2(2,s2)))
F1(M(2,kg), B(L1(-6,mm), T2(2,s2)))
G(M(-3.75,kg) * (B(L1(1,m), T2(1,s2))))
F1(M(7.5,kg), B(L1(1,m), T2(1,s2)))
>>> f1.to(cgs)
"-200.0 ['g', 'cm']/['s', 's']"
>>> f1.to(mks)
"-0.0 ['kg', 'm']/['s', 's']"
>>> f1.to(kmh)
"-25.92 ['kg', 'km']/['h', 'h']"
>>> f1.to(mph)
"-16.11 ['kg', 'mi']/['h', 'h']"
>>> f2.to(kmh)
"-77.76 ['kg', 'km']/['h', 'h']"
>>> f3.to(cgs)
"-375000.0 ['g', 'cm']/['s', 's']"
>>> f4.to(mks)

```

```

"7.5 ['kg', 'm']/['s', 's']"
>>> newton(330).to(cgs) # globale Methode
"33000000.0 ['g', 'cm']/['s', 's']"
>>> f1.to(hkm)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 2144, in to
    raise ValueError(_fehler5 + str(d))
ValueError: Parameter ist unzulässig: hkm

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

15 Abgeleitete Klasse W

W ist eine Subklasse von G und erbt daher auch alle Eigenschaften (Abschnitt 9.5 auf Seite 89) und Methoden (Abschnitt 9.2 auf Seite 82) von G (Abschnitt 9 auf Seite 81). Sie realisiert das Rechnen mit Arbeit:

$$W = \text{Kraft} \times \text{Länge} = \text{Masse} \times \text{Länge} \times \text{Beschleunigung} \left[\text{kg} \times \frac{\text{m}^2}{\text{s}^2} \right]$$

15.1 Übersicht

Globale Informationen über die Klasse W

Eine Übersicht über die Klasse W erhalten Sie durch die globale Methode `dok("W")` oder – etwas ausführlicher – mit der W-Methode `W.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("W.classInfo")`:

```
>>> dok("W.classInfo")
W.classInfo
    Gibt Informationen zur Klasse W und ihren Methoden aus.

    Aufruf: W.classInfo(m=art) oder W.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> W.classInfo("H")
Class W

    W ist eine Subklasse von G und realisiert das Rechnen mit W-Instanzen
    (Arbeit). Sie erbt damit alle Eigenschaften und Methoden von G (außer
    denen, die überschrieben werden).

    Aufruf: W.__init__(k, l, n="")

    Parameter:

    k      : K-Instanz;           keine Voreinstellung
```

15 Abgeleitete Klasse W

`l` : L1-Instanz; keine Voreinstellung
`n=""` : Name der W-Instanz; Voreinstellung: leere Zeichenkette

Methoden:

`w` : W-Instanz (Arbeit)
`e` : eine Methode der Klasse W
`o` : eine zulässige Instanz

<code>w.__abs__()</code> oder <code>abs(w)</code>	Liefert den Absolut-Betrag der W-Instanz.
<code>w.__add__(o)</code> oder <code>w + o</code>	Realisiert das Addieren von W-Instanzen.
<code>W.__init__(k, l, n='')</code>	Initialisiert eine W-Instanz.
<code>w.__mul__(o)</code> oder <code>w * o</code>	Realisiert das Multiplizieren von W-Instanzen.
<code>w.__repr__()</code> oder <code>repr(w)</code>	Repräsentiert eine W-Instanz.
<code>w.__str__()</code> oder <code>str(w)</code>	Repräsentiert eine W-Instanz.
<code>w.__sub__(o)</code> oder <code>w - o</code>	Realisiert das Subtrahieren von W-Instanzen.
<code>w.__eq__(w)</code> oder <code>w == w</code>	Realisiert den Vergleich <code>==</code> in W.
<code>w.__ge__(w)</code> oder <code>w >= w</code>	Realisiert den Vergleich <code>>=</code> in W.
<code>w.__gt__(w)</code> oder <code>w > w</code>	Realisiert den Vergleich <code>></code> in W.
<code>w.__le__(w)</code> oder <code>w <= w</code>	Realisiert den Vergleich <code><=</code> in W.
<code>w.__lt__(w)</code> oder <code>w < w</code>	Realisiert den Vergleich <code><</code> in W.
<code>w.__ne__(w)</code> oder <code>w != w</code>	Realisiert den Vergleich <code>!=</code> in W.
<code>W.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse W.
<code>W.classInfo(m=art)</code>	Gibt Informationen zur Klasse W und ihren Methoden aus.
<code>W.ci(m=art)</code>	Alias für <code>W.classInfo(m=art)</code>
<code>W.description()</code>	Liefert eine Kurzbeschreibung der Klasse W.
<code>w.to(liste)</code>	Gibt in geeigneter Weise den Wert einer W-Instanz aus.
<code>w.info()</code>	Gibt Informationen über eine W-Instanz (Arbeit) aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt [15.2](#)
- Maßeinheiten und Instanzen → Abschnitt [15.3 auf der nächsten Seite](#)
- Eigenschaften der Instanzen → Abschnitt [15.5 auf Seite 142](#)
- Zulässige Operationen → Abschnitt [15.4 auf Seite 140](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [15.6 auf Seite 143](#)

15.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("W")` auf der vorherigen Seite liefert u. a. eine Auflistung der W-Methoden. Namen, die die Zeichenfolge `__` enthalten, stehen für interne Methoden der

Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 15.1 aufgeführt.

Tabelle 15.1: Methoden der Klasse *W*

Name	Bedeutung
<code>W.beispiel(<i>e</i>)</code>	Liefert Beispiele für die Methoden der Klasse <i>W</i> .
<code>W.classInfo(<i>art</i>)</code>	Gibt Informationen zur Klasse <i>W</i> und ihren Methoden aus.
<code>W.ci(<i>art</i>)</code>	Alias für <code>classInfo(<i>art</i>)</code>
<code>W.description()</code>	Liefert eine einzeilige Kurzbeschreibung der Klasse <i>W</i> .
<code><i>w</i>.info(<i>modus</i>)</code>	Gibt Informationen über die <i>W</i> -Instanz <i>w</i> (Arbeit) aus.

Ausführlichere Informationen über eine bestimmte *W*-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `W.classInfo("M")` für alle *W*-Methoden.

Beispiele für `dok`

```
>>> dok("W.to")
W.to
    Gibt in geeigneter Weise den Wert einer G-Instanz aus.

    Aufruf: g.to(wliste)
    vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
    - leer : keine Konvertierung [Voreinstellung], aber Bereinigung
    - kmh  : m --> km, s --> h
    - mks  : m --> m, s --> s, kg --> kg
    - cgs  : m --> cm, kg --> g, s --> s
    - mph  : m --> mi, s --> h
    mögliche Fehlermeldung:
    - Parameter ist unzulässig.

>>> dok("W.info")
W.info
    Gibt Informationen über eine W-Instanz (Arbeit) aus.

    Aufruf: w.info()
```

15.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse *W* besitzt keine eigenen Maßeinheiten; sie nutzt die von *M* (Abschnitt 7.3 auf Seite 66), *L1* (Abschnitt 2.3 auf Seite 18) und *T2* (Abschnitt 6.3 auf Seite 57).

Instanzen

Eigene W-Instanzen können Sie mittels der Anweisung $W(kraft, länge, n=name)$ oder der globalen Methode `joule(wert)` (beide Arbeit) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> f1 = newton(100)                # globale Methode
>>> f1
F1(M(100,kg), B(L1(1,m), T2(1,s2)))
>>> w1 = joule(-200)                # globale Methode
>>> w2 = W(f1, 111)
>>> w3 = +joule(3.75)               # globale Methode
>>> w4 = W(f1, 111 * 2, n="doppelt")
>>> w5 = -joule(200)
>>> w1
W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> w2
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(2,m))
>>> w3
W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> w4
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(4,m))
>>> w5
G(F1(M(-200.0,kg), B(L1(1,m), T2(1,s2))) * (L1(1,m)))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene W-Instanz zu generieren, kann misslingen:

- ☐ **Operand hat keinen zulässigen Typ:** Der erste Parameter *kraft* muss vom Typ F1, der zweite *länge* vom Typ L1 sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("W.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne W-Instanz erhalten Sie mit Hilfe der W-Methode `info`, beispielsweise für `w3`:

```
>>> w3.info()
Name           : joule(3.75)
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(3.75,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand: L1(1,m)
Wert           : 3.75 (in den angegebenen Maßeinheiten)
interner Wert  : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
```


15 Abgeleitete Klasse W

```
konvertiert (cgs)      : 37500000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 3.75 ['kg', 'm', 'm']/['s', 's']
-----
```

Informationen über alle W-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("W")
5 Element(e):
```

```
Name          : joule(-200)
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(-200,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand: L1(1,m)
Wert           : -200.0 (in den angegebenen Maßeinheiten)
interner Wert  : -200.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
konvertiert (cgs)      : -2000000000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: -200.0 ['kg', 'm', 'm']/['s', 's']
-----
```

```
Name          : W(L1(2,m), t)
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(100,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand: L1(2,m)
Wert           : 200.0 (in den angegebenen Maßeinheiten)
interner Wert  : 200.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
konvertiert (cgs)      : 2000000000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 200.0 ['kg', 'm', 'm']/['s', 's']
-----
```

```
Name          : joule(3.75)
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(3.75,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand: L1(1,m)
Wert           : 3.75 (in den angegebenen Maßeinheiten)
interner Wert  : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
konvertiert (cgs)      : 37500000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 3.75 ['kg', 'm', 'm']/['s', 's']
-----
```

```
Name          : doppelt
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(100,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand : L1(4,m)
Wert           : 400.0 (in den angegebenen Maßeinheiten)
interner Wert   : 400.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
konvertiert (cgs)      : 4000000000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 400.0 ['kg', 'm', 'm']/['s', 's']
-----
```

```
Name          : joule(200)
Art            : W-Instanz (Arbeit)
linker Operand : F1(M(200,kg), B(L1(1,m), T2(1,s2)))
Operator       : *
rechter Operand : L1(1,m)
Wert           : 200.0 (in den angegebenen Maßeinheiten)
interner Wert   : 200.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner  : ['s2']
konvertiert (cgs)      : 2000000000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 200.0 ['kg', 'm', 'm']/['s', 's']
-----
```

15.4 Operatoren

Vergleiche

Instanzen der Klasse W können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«
4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```
>>> w1; w2; w3; w4
W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(2,m))
W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(4,m))
```

```

>>> w1 <= w2
True
>>> w3 >= w4
False
>>> w2 != joule(23)    # globale Methode
True
>>> w4 == joule(3.4)   # globale Methode
False

```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```

>>> w1
W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> (joule(1) <= w1) and (w1 <= joule(5))    # t11 zwischen 1J und 5J
False
>>> not ((w1 == joule(1)) or (w1 == joule(5))) # t11 weder 1J noch 5J
True

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse W verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 15.2.

Tabelle 15.2: Klasse W, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	W	G	G	G	G	G	G	W
/	G	L1	G	F1	G	G	G	W	G	P	G	G	G	Skalar	W
+	–	–	–	–	–	–	–	W	–	–	–	–	–	–	–
-	–	–	–	–	–	–	–	W	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

Beispiele – arithmetische Operatoren

```

>>> w1; w2; w3
W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(2,m))
W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> w1 + w2
W(F1(M(0.0,kg), B(L1(1,m), T2(1,s2))), L1(1,m))

```

```

>>> w3 - joule(23) # globale Methode
W(F1(M(-19.25,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> joule(3.4) * 3 # globale Methode
W(F1(M(10.2,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> w1 / w2
-1.0
>>> w3 ** 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'W' and 'int'

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit W-Instanzen erhalten Sie durch die Methode `W.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("W.beispiel")`, beispielsweise `W.beispiel()`.

15.5 Eigenschaften der W-Instanzen

Jede W-Instanz (jedes W-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der W-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `W.classInfo("V")`. Danach stehen in der Klasse W die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> W.classInfo("V")
Eigenschaften der W-Instanzen:

w.links      linker Operand
w.operator   Operator
w.rechts     rechter Operand
w.name       Name der Instanz
w.oben       Einheiten im Zähler
w.unten      Einheiten im Nenner
w.internal   interner Wert
w.v          Wert in den angegebenen Einheiten

```

Beispiele

```

>>> w4
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(4,m))
>>> w4.info()
Name           : doppelt
Art            : W-Instanz (Arbeit)

```

```

linker Operand      : F1(M(100,kg), B(L1(1,m), T2(1,s2)))
Operator            : *
rechter Operand     : L1(4,m)
Wert                : 400.0 (in den angegebenen Maßeinheiten)
interner Wert       : 400.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2']
konvertiert (cgs)    : 4000000000.0 ['g', 'cm', 'cm']/['s', 's']
konvertiert (mks) [Nm=J]: 400.0 ['kg', 'm', 'm']/['s', 's']
-----

```

```

>>> w4.links
F1(M(100,kg), B(L1(1,m), T2(1,s2)))
>>> w4.operator
'*'
>>> w4.rechts
L1(4,m)
>>> w4.name
'doppelt'
>>> w4.oben
['kg', 'm', 'm']
>>> w4.unten
['s2']
>>> w4.internal
400.0
>>> w4.v
400.0

```

15.6 Konvertierung

Instanzen der Klasse W können mit der Hilfe der W-Methode `to(einheitensystem)` in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

leer : keine Konvertierung, aber Bereinigung

kmh : m \longrightarrow km, s \longrightarrow h

mks : m \longrightarrow m, s \longrightarrow s, kg \longrightarrow kg

cgs : m \longrightarrow cm, kg \longrightarrow g, s \longrightarrow s

mph : m \longrightarrow mi, s \longrightarrow h

Beispiele – Konvertierung

```

>>> w1; w2; w3; w4
W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(2,m))
W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
W(F1(M(100,kg), B(L1(1,m), T2(1,s2))), L1(4,m))

```

```

>>> w1.to(cgs)
"-2000000000.0 ['g', 'cm', 'cm']/['s', 's']"
>>> w1.to(mks)
"-200.0 ['kg', 'm', 'm']/['s', 's']"
>>> w1.to(kmh)
"-2592.0 ['kg', 'km', 'km']/['h', 'h']"
>>> w1.to(mph)
"-1000.78 ['kg', 'mi', 'mi']/['h', 'h']"
>>> w2.to(kmh)
"2592.0 ['kg', 'km', 'km']/['h', 'h']"
>>> w3.to(cgs)
"37500000.0 ['g', 'cm', 'cm']/['s', 's']"
>>> joule(300).to(cgs) # globale Methode
"3000000000.0 ['g', 'cm', 'cm']/['s', 's']"
>>> w4.to(hkm)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 2144, in to
    raise ValueError(_fehler5 + str(d))
ValueError: Parameter ist unzulässig: hkm

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

16 Abgeleitete Klasse P

P ist eine Subklasse von G und erbt daher auch alle Eigenschaften (Abschnitt 9.5 auf Seite 89) und Methoden (Abschnitt 9.2 auf Seite 82) von G (Abschnitt 9 auf Seite 81). Sie realisiert das Rechnen mit mechanischen Leistungen:

$$P = \frac{\text{Arbeit}}{\text{Zeit}} = \frac{\text{Kraft} \times \text{Länge}}{\text{Zeit}} \left[\frac{\text{kg} \times \text{m}^2}{\text{s}^3} \right]$$

16.1 Übersicht

Globale Informationen über die Klasse P

Eine Übersicht über die Klasse P erhalten Sie durch die globale Methode `dok("P")` oder – etwas ausführlicher – mit der P-Methode `P.classInfo(ausgabe)`. Welche Angaben für *ausgabe* möglich sind, erfahren Sie durch `dok("P.classInfo")`:

```
>>> dok("P.classInfo")
P.classInfo
    Gibt Informationen zur Klasse P und ihren Methoden aus.

    Aufruf: P.classInfo(m=art) oder P.ci(m=art)
    mögliche Angaben für m:
    + "A"/alles      : alles [Voreinstellung]
    + "H"/Kopf       : globale Informationen
    + "V"/Variablen  : Variablen/Eigenschaften
    + "M"/Methoden   : Methoden
```

Damit erhalten Sie u. a. folgende Informationen:

```
>>> P.classInfo("H")
Class P

    P ist eine Subklasse von G und realisiert das Rechnen mit P-Instanzen
    (Leistung). Sie erbt damit alle Eigenschaften und Methoden von G (außer
    denen, die überschrieben werden).

    Aufruf: P.__init__(w, t, n="")

    Parameter:
```

16 Abgeleitete Klasse P

w : W-Instanz; keine Voreinstellung
t : T1-Instanz; keine Voreinstellung
n="" : Name der P-Instanz; Voreinstellung: leere Zeichenkette

Methoden:

p : P-Instanz (Leistung)
e : eine Methode der Klasse P
o : ein zulässige Instanz

p.__abs__() oder abs(p)	Liefert den Absolut-Betrag der P-Instanz.
p.__add__(o) oder p + o	Realisiert das Addieren von P-Instanzen.
P.__init__(w, t, n='')	Initialisiert eine P-Instanz.
p.__mul__(o) oder p * o	Realisiert das Multiplizieren von P-Instanzen.
p.__repr__() oder repr(p)	Repräsentiert eine P-Instanz.
p.__str__() oder str(p)	Repräsentiert eine P-Instanz.
p.__sub__(o) oder p - o	Realisiert das Subtrahieren von P-Instanzen.
p.__truediv__(o) oder p / o	Realisiert die Division von P-Instanzen.
p.__eq__(p) oder p == p	Realisiert den Vergleich == in P.
p.__ge__(p) oder p >= p	Realisiert den Vergleich >= in P.
p.__gt__(p) oder p > p	Realisiert den Vergleich > in P.
p.__le__(p) oder p <= p	Realisiert den Vergleich <= in P.
p.__lt__(p) oder p < p	Realisiert den Vergleich < in P.
p.__ne__(p) oder p != p	Realisiert den Vergleich != in P.
P.beispiel(e)	Liefert Beispiele für die Methoden der Klasse P.
P.classInfo(m=art)	Gibt Informationen zur Klasse P und ihren Methoden aus.
P.ci(m=art)	Alias für P.classInfo(m=art)
P.description()	Liefert eine Kurzbeschreibung der Klasse P.
p.to(liste)	Gibt in geeigneter Weise den Wert einer P-Instanz aus.
p.info()	Gibt Informationen über eine P-Instanz (Leistung) aus.

Weitere Informationen

- Methoden der Klasse → Abschnitt [16.2 auf der nächsten Seite](#)
- Maßeinheiten und Instanzen → Abschnitt [16.3 auf Seite 148](#)
- Eigenschaften der Instanzen → Abschnitt [16.5 auf Seite 152](#)
- Zulässige Operationen → Abschnitt [16.4 auf Seite 150](#)
- Konvertierung in andere zulässige Maßeinheiten → Abschnitt [16.6 auf Seite 153](#)

16.2 Methoden der Klasse

Der Aufruf der globalen Methode `dok("P")` auf Seite 145 liefert u. a. eine Auflistung der P-Methoden. Namen, die die Zeichenfolge `--` enthalten, stehen für interne Methoden der Klasse; sie sollten nicht direkt verwendet werden. Die für Endnutzer gedachten Methoden sind in Tabelle 16.1 aufgeführt.

Tabelle 16.1: Methoden der Klasse P

Name	Bedeutung
<code>P.beispiel(e)</code>	Liefert Beispiele für die Methoden der Klasse P.
<code>P.classInfo(art)</code>	Gibt Informationen zur Klasse P und ihren Methoden aus.
<code>P.ci(art)</code>	Alias für <code>classInfo(art)</code>
<code>P.description()</code>	Liefert eine einzeilige Kurzbeschreibung der Klasse P.
<code>p.info(modus)</code>	Gibt Informationen über die P-Instanz <i>p</i> (Leistung) aus.

Ausführlichere Informationen über eine bestimmte P-Methode erhalten Sie durch die globale Methode `dok(name)` oder durch `P.classInfo("M")` für alle P-Methoden.

Beispiele für dok

```
>>> dok("P.to")
```

```
P.to
```

```
Gibt in geeigneter Weise den Wert einer G-Instanz aus.
```

```
Aufruf: g.to(wliste)
```

```
vorgefertigte Werte für wliste (Dictionary mit Konvertierungen):
```

```
- leer : keine Konvertierung [Voreinstellung], aber Bereinigung
```

```
- kmh : m --> km, s --> h
```

```
- mks : m --> m, s --> s, kg --> kg
```

```
- cgs : m --> cm, kg --> g, s --> s
```

```
- mph : m --> mi, s --> h
```

```
mögliche Fehlermeldung:
```

```
- Parameter ist unzulässig.
```

```
>>> dok("P.info")
```

```
P.info
```

```
Gibt Informationen über eine P-Instanz (Leistung) aus.
```

```
Aufruf: p.info()
```

16.3 Maßeinheiten und Instanzen

Maßeinheiten

Die Klasse P besitzt keine eigenen Maßeinheiten; sie nutzt die von M (Abschnitt 7.3 auf Seite 66), L2 (Abschnitt 3.3 auf Seite 29), T1 (Abschnitt 5.3 auf Seite 48) und T2 (Abschnitt 6.3 auf Seite 57).

Instanzen

Eigene P-Instanzen können Sie mittels der Anweisung `P(arbeit,zeit,n=name)` oder der globalen Methoden `watt(wert)` und `PS(wert)` (beide Leistung) generieren. Gegebenenfalls können Sie ein Vorzeichen (+ oder -) voranstellen:

```
>>> w1 = joule(1000) # globale Methode
>>> p0 = watt(-200)  # globale Methode
>>> t11
T1(2,s)
>>> p1 = P(w1, t11)
>>> p2 = +watt(3.75) # globale Methode
>>> p3 = PS(75)      # globale Methode
>>> p4 = -watt(350)  # globale Methode
>>> p0
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2)))), L1(1,m)), T1(1,s))
>>> p1
P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2)))), L1(1,m)), T1(2,s))
>>> p2
P(W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2)))), L1(1,m)), T1(1,s))
>>> p3
P(W(F1(M(55162.41,kg), B(L1(1,m), T2(1,s2)))), L1(1,m)), T1(1,s))
>>> p4
G(W(F1(M(-350.0,kg), B(L1(1,m), T2(1,s2)))), L1(1,m)) / (T1(1,s)))
```

Mögliche Fehlerquellen

Der Versuch, eine eigene P-Instanz zu generieren, kann misslingen:

- **Operand hat den Wert Null:** Der zweite Parameter *zeit* darf nicht den Wert Null haben.
- **Operand hat keinen zulässigen Typ:** Der erste Parameter *arbeit* muss vom Typ W, der zweite *zeit* vom Typ T1 sein.

Näheres erfahren Sie mit einem Aufruf der globalen Methode `dok("P.__init__")`.

Informationen über Instanzen

Informationen über eine einzelne P-Instanz erhalten Sie mit Hilfe der P-Methode `info`, beispielsweise für `p1`:

16 Abgeleitete Klasse P

```
>>> p1.info()
Name          : P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
Art           : P-Instanz (Leistung)
linker Operand : W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator      : /
rechter Operand : T1(2,s)
Wert          : 500.0 (in den angegebenen Maßeinheiten)
interner Wert  : 500.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 5000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 500.0 ['kg', 'm', 'm']/['s', 's', 's']
-----
```

Informationen über alle P-Instanzen erhalten Sie mittels der globalen Methode `alle`:

```
>>> alle("P")
5 Element(e):

Name          : watt(-200)
Art           : P-Instanz (Leistung)
linker Operand : W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator      : /
rechter Operand : T1(1,s)
Wert          : -200.0 (in den angegebenen Maßeinheiten)
interner Wert  : -200.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : -2000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : -200.0 ['kg', 'm', 'm']/['s', 's', 's']
-----

Name          : P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
Art           : P-Instanz (Leistung)
linker Operand : W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator      : /
rechter Operand : T1(2,s)
Wert          : 500.0 (in den angegebenen Maßeinheiten)
interner Wert  : 500.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 5000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 500.0 ['kg', 'm', 'm']/['s', 's', 's']
-----

Name          : watt(3.75)
Art           : P-Instanz (Leistung)
```

16 Abgeleitete Klasse P

```
linker Operand      : W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator            : /
rechter Operand     : T1(1,s)
Wert                : 3.75 (in den angegebenen Maßeinheiten)
interner Wert       : 3.75 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 37500000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 3.75 ['kg', 'm', 'm']/['s', 's', 's']
-----

Name                : watt(55162.41)
Art                 : P-Instanz (Leistung)
linker Operand      : W(F1(M(55162.41,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator            : /
rechter Operand     : T1(1,s)
Wert                : 55162.41 (in den angegebenen Maßeinheiten)
interner Wert       : 55162.41 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 551624062500.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 55162.41 ['kg', 'm', 'm']/['s', 's', 's']
-----

Name                : watt(350)
Art                 : P-Instanz (Leistung)
linker Operand      : W(F1(M(350,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator            : /
rechter Operand     : T1(1,s)
Wert                : 350.0 (in den angegebenen Maßeinheiten)
interner Wert       : 350.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 3500000000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 350.0 ['kg', 'm', 'm']/['s', 's', 's']
-----
```

16.4 Operatoren

Vergleiche

Instanzen der Klasse P können mit den folgenden Vergleichsoperatoren verglichen werden:

1. < »kleiner«
2. > »größer«
3. <= »kleiner oder gleich«

4. >= »größer oder gleich«
5. == »gleich«
6. != »ungleich«

Beispiele – Vergleiche

```
>>> p0; p1; p2
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
P(W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> p0 <= p1
True
>>> p1 >= p2
True
>>> p1 != watt(23) # globale Methode
True
>>> p2 == p0
False
```

Die Vergleiche sind nur dann zulässig, wenn die beiden Operanden vom gleichen Typ sind. Bei unzulässigen Vergleichen wird eine Fehlermeldung ausgegeben.

Die obigen logischen Ausdrücke können geklammert und mit den logischen Operatoren **and**, **or** oder **not** verknüpft werden:

Beispiele – logische Ausdrücke

```
>>> p0
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> (watt(1) <= p0) and (p0 <= watt(5)) # p0 zwischen 1W und 5W
False
>>> not ((p0 == watt(1)) or (p0 == watt(5))) # p0 weder 1W noch 5W
True
```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Arithmetische Operatoren

Die Klasse P verfügt über die arithmetischen Operatoren +, -, *, / und ** (mit Einschränkungen). Siehe dazu Tabelle 16.2 auf der nächsten Seite.

Beispiele – arithmetische Operatoren

```
>>> p0; p1; p2
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
P(W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> p0 + p1
P(W(F1(M(300.0,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> p2 - watt(23) # globale Methode
```

Tabelle 16.2: Klasse P, arithmetische Operatoren und rechtsseitige Operanden

	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
*	G	G	G	G	G	G	G	P	G	W	G	G	G	G	P
/	G	V	G	G	G	G	G	P	Skalar	G	G	G	F1	G	P
+	–	–	–	–	–	–	–	P	–	–	–	–	–	–	–
–	–	–	–	–	–	–	–	P	–	–	–	–	–	–	–
**	–	–	–	–	–	–	–	?	–	–	–	–	–	–	?

```

P(W(F1(M(-19.25,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> watt(3.4) * 2 # globale Methode
P(W(F1(M(6.8,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> p0 / p2
-53.333333333333336
>>> p2 ** 2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for ** or pow(): 'P' and 'int'

```

Bei unzulässigen Operationen wird eine Fehlermeldung ausgegeben.

Beispiele

Beispiele für den Umgang mit P-Instanzen erhalten Sie durch die lokale Methode P-Methode `P.beispiel(beispiel)`. Welche Angaben Sie für *beispiel* machen können, erfahren Sie durch die globale Methode `dok("P.beispiel")`, beispielsweise `P.beispiel()`.

16.5 Eigenschaften der P-Instanzen

Jede P-Instanz (jedes P-Objekt) ist durch verschiedene Eigenschaften gekennzeichnet. Durch einen Aufruf der P-Methode `info` können Sie sich diese auflisten lassen. Sie können aber auch gezielt auf bestimmte Eigenschaften zugreifen. Welche das sind, erfahren Sie durch einen Aufruf der Methode `P.classInfo("V")`. Danach stehen in der Klasse P die folgenden Eigenschaften für Instanzen zur Verfügung:

```

>>> P.classInfo("V")
Eigenschaften der P-Instanzen:

p.links      linker Operand
p.operator   Operator
p.rechts     rechter Operand
p.name       Name der Instanz
p.oben       Einheiten im Zähler
p.unten      Einheiten im Nenner
p.internal   interner Wert

```

p.v Wert in den angegebenen Einheiten

Beispiele

```
>>> p1
P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
>>> p1.info()
Name          : P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
Art           : P-Instanz (Leistung)
linker Operand : W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
Operator      : /
rechter Operand : T1(2,s)
Wert          : 500.0 (in den angegebenen Maßeinheiten)
interner Wert  : 500.0 (in den Basismaßeinheiten [m, kg, s, K])
Maßeinheit(en) Zähler : ['kg', 'm', 'm']
Maßeinheit(en) Nenner : ['s2', 's']
konvertiert (cgs)   : 5000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']
konvertiert (mks) [W] : 500.0 ['kg', 'm', 'm']/['s', 's', 's']
-----

>>> p1.links
W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> p1.operator
 '/'
>>> p1.rechts
T1(2,s)
>>> p1.name
'P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))'
>>> p1.oben
['kg', 'm', 'm']
>>> p1.unten
['s2', 's']
>>> p1.internal
500.0
>>> p1.v
500.0
```

16.6 Konvertierungen

Instanzen der Klasse P können mit der Hilfe der P-Methode `to(einheitensystem)` in andere Einheitensysteme umgerechnet werden. Für *einheitensystem* sind folgende Angaben möglich:

leer : keine Konvertierung, aber Bereinigung
`kmh` : m \longrightarrow km, s \longrightarrow h
`mks` : m \longrightarrow m, s \longrightarrow s, kg \longrightarrow kg

$\text{cgs} : \text{m} \longrightarrow \text{cm}, \text{kg} \longrightarrow \text{g}, \text{s} \longrightarrow \text{s}$
 $\text{mph} : \text{m} \longrightarrow \text{mi}, \text{s} \longrightarrow \text{h}$

Beispiele – Konvertierung

```

>>> p0; p1; p2; p3
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
P(W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(2,s))
P(W(F1(M(3.75,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
P(W(F1(M(55162.41,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
>>> p0.to(cgs)
"-2000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']"
>>> p0.to(mks)
"-200.0 ['kg', 'm', 'm']/['s', 's', 's']"
>>> p0.to(kmh)
"-9331200.0 ['kg', 'km', 'km']/['h', 'h', 'h']"
>>> p0.to(mph)
"-3602796.46 ['kg', 'mi', 'mi']/['h', 'h', 'h']"
>>> p1.to(kmh)
"23328000.0 ['kg', 'km', 'km']/['h', 'h', 'h']"
>>> p2.to(cgs)
"37500000.0 ['g', 'cm', 'cm']/['s', 's', 's']"
>>> watt(300).to(cgs) # globale Methode
"30000000000.0 ['g', 'cm', 'cm']/['s', 's', 's']"
>>> p3.to(mks)
"55162.41 ['kg', 'm', 'm']/['s', 's', 's']"
>>> p1.to(hkm)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "D:\Python\units.py", line 2144, in to
    raise ValueError(_fehler5 + str(d))
ValueError: Parameter ist unzulässig: hkm

```

Der Versuch, in eine unzulässige Maßeinheit zu konvertieren, führt zu einer Fehlermeldung.

17 Kombinierte Anwendungen

17.1 Kombinierte Operationen

Instanzen der verschiedenen Klassen können mit Hilfe der Operatoren $*$ und $/$ kombiniert werden. Voraussetzung ist, dass die Klasse des linken Operanden eine solche Operation zulässt. Siehe dazu Tabelle 17.1 und Tabelle 17.2.

Tabelle 17.1: Zulässige Operationen mit dem Operator $*$

*	rechter Operand														
	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Skalar
B	G	G	G	G	G	G	F1	B	G	V	L1	G	G	G	B
F1	G	G	G	W	G	G	G	F1	G	G	G	G	P	G	F1
G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
L1	G	G	G	L2	L3	G	G	L1	G	G	G	G	G	G	L1
L2	G	G	G	L3	G	G	G	L2	G	G	G	G	G	G	L2
L3	G	G	G	G	G	G	G	L3	G	G	G	G	G	G	L3
M	F1	G	G	G	G	G	G	M	G	G	G	G	G	G	M
N	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	N
P	G	G	G	G	G	G	G	P	G	W	G	G	G	G	P
T1	V	G	G	G	G	G	G	T1	W	T2	G	G	G	G	T1
T2	G	G	G	G	G	G	G	T2	G	G	G	G	G	G	T2
TT	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(TT)	(G)	(G)	(TT)
U	—	—	—	—	—	—	—	(U)	—	—	—	—	—	—	(U)
V	G	G	G	G	G	G	G	G	G	L1	G	G	G	G	V
W	G	G	G	G	G	G	G	W	G	G	G	G	G	G	W

Tabelle 17.2: Zulässige Operationen mit dem Operator $/$

/	rechter Operand														
	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Sk.
B	Sk.	G	G	G	G	G	G	B	G	G	G	G	G	G	B
F1	M	Sk.	G	G	G	G	B	F1	G	G	G	G	G	G	F1
G	G	G	G	G	G	G	G	G	G	G	G	G	G	G	G
L1	T2	G	G	Sk.	G	G	G	L1	G	V	B	G	T1	G	L1
L2	G	G	G	L1	Sk.	G	G	L2	G	G	G	G	G	G	L2
L3	G	G	G	L2	L1	Sk.	G	L3	G	G	G	G	G	G	L3
M	G	G	G	G	G	G	Sk.	M	G	G	G	G	G	G	M

(Fortsetzung auf nächster Seite)

(Fortsetzung)

/	rechter Operand														
	B	F1	G	L1	L2	L3	M	N	P	T1	T2	TT	V	W	Sk.
N	G	G	G	G	G	G	G	N	G	G	G	G	G	G	N
P	G	V	G	G	G	G	G	P	Sk.	G	G	G	F1	G	P
T1	G	G	G	G	G	G	G	T1	G	Sk.	G	G	G	G	T1
T2	G	G	G	G	G	G	G	T2	G	T1	Sk.	G	G	G	T2
TT	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(G)	(Sk.)	(G)	(G)	(TT)
U	—	—	—	—	—	—	—	(U)	—	—	—	—	—	—	(U)
V	T1	G	G	G	G	G	G	V	G	B	G	G	Sk.	G	V
W	G	L1	G	F1	G	G	G	W	G	P	G	G	G	Sk.	W

Beispiele

```

>>> l11; l21; v0; f1; t21
L1(2,m)
L2(2,m2)
V(L1(3,m), T1(1,s))
F1(M(100,kg), B(L1(1,m), T2(1,s2)))
T2(2,s2)
>>> l11/v0      # L1/V  --> T1
T1(0.67,s)
>>> l11*f1      # L1*F1 --> W (Arbeit)
W(F1(M(200.0,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
>>> l11/t21     # L1/T2 --> B (Beschleunigung)
B(L1(1.0,m), T2(1,s2))
>>> l21/l11     # L2/L1 --> L1
L1(1.0,m)
>>> v0/t11      # V*T1  --> B (Beschleunigung)
B(L1(1.5,m), T2(1,s2))

>>> p0; m11; w1; b1
P(W(F1(M(-200,kg), B(L1(1,m), T2(1,s2))), L1(1,m)), T1(1,s))
M(2,kg)
W(F1(M(1000,kg), B(L1(1,m), T2(1,s2))), L1(1,m))
B(L1(-2,mm), T2(2,s2))
>>> p0/v0      # P/V  --> F1 (Kraft)
F1(M(-66.67,kg), B(L1(1,m), T2(1,s2)))
>>> m11*b1     # M*B   --> F1 (Kraft)
F1(M(-0.0,kg), B(L1(1,m), T2(1,s2)))
>>> f1/m11     # F1/M  --> B (Beschleunigung)
B(L1(50.0,m), T2(1,s2))
>>> w1/l11     # W/L1  --> F1 (Kraft)
F1(M(500.0,kg), B(L1(1,m), T2(1,s2)))
>>> w1/f1      # W/F1  --> L1
L1(10.0,m)

```

17.2 Globale Methoden

Das Modul 'units.py' (in der Version '9.16.2' vom '2018-07-29') stellt einige globale Methoden, in denen Klassen kombiniert werden, zur Verfügung. Siehe Tabelle [1.3 auf Seite 13](#).

Beschreibungen erhalten Sie beispielsweise durch

```
>>> dok("druck")
druck
    Anwendung: Berechnet (=Kraft/Fläche).

    Aufruf: druck(k, f)
    k : F1-Instanz (Kraft)
    f : L2-Instanz (Fläche)
    mögliche Fehlermeldungen:
    - Operand hat keinen zulässigen Typ.
    - Operand hat den Wert Null.
```

Beispiele

```
>>> l11, l21, l31, t11
(L1(2,m), L2(2,m2), L3(2,m3), T1(2,s))
>>> f1; m11; v0
F1(M(100,kg), B(L1(1,m), T2(1,s2)))
M(2,kg)
V(L1(3,m), T1(1,s))

>>> druck(f1, l21)
"50.0 ['kg']/['m', 's', 's']"
>>> gprocm3(m11, l31)
"0.0 ['g']/['cm3']"
>>> impuls(m11, v0)
"6.0 ['kg', 'm']/['s']"
>>> literpro100km(l31, l11)
"100000000.0 ['liter']/['hkm']"
>>> literprom2(l31, l21)
"1000.0 ['liter']/['m2']"
>>> mpros(l11, t11)
"1.0 ['m']/['s']"
>>> mgprodl(m11, l31)
"100.0 ['mg']/['dl']"
>>> Upromin(200, t11)
"6000.0 []/['minute']"
>>> frequenz(3000)
G(N(3000) / (T1(1,s)))
```

Index

- **** (Operator), 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 111, 112, 121, 122, 131, 141, 142, 151, 152
- *** (Operator), 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 111, 112, 121, 122, 131, 141, 142, 151, 152, 155
- +** (Operator), 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 79, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152
- +** (Vorzeichen), 19, 20, 29, 30, 39, 40, 48, 57, 67, 83, 84, 93, 100, 108, 118, 128, 138, 148
- (Operator), 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 79, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152
- (Vorzeichen), 19, 20, 29, 30, 39, 40, 48, 49, 57, 67, 83, 84, 93, 100, 108, 118, 128, 138, 148
- /** (Operator), 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152, 155, 156
- <=** (Vergleich), 5, 22, 23, 31, 32, 41, 42, 50, 51, 59, 60, 69, 70, 77, 78, 87, 94, 95, 101, 102, 111, 120, 121, 130, 131, 140, 141, 150, 151
- <** (Vergleich), 5, 22, 31, 41, 50, 59, 69, 77, 87, 94, 101, 110, 120, 130, 140, 150
- ==** (Vergleich), 5, 22, 23, 31, 32, 41, 42, 50, 51, 59, 60, 69, 70, 77, 78, 87, 88, 94, 95, 101, 102, 111, 120, 121, 130, 131, 140, 141, 151
- >=** (Vergleich), 5, 22, 31, 32, 41, 42, 50, 51, 59, 60, 69, 77, 87, 94, 101, 111, 120, 121, 130, 140, 141, 151
- >** (Vergleich), 5, 22, 31, 41, 50, 59, 69, 77, 87, 94, 101, 111, 120, 130, 140, 150
- !=** (Vergleich), 22, 31, 32, 41, 42, 50, 51, 59, 60, 69, 77, 78, 87, 94, 101, 102, 111, 120, 121, 130, 140, 141, 151
- alle** (globale Methode), 8, 11, 13, 21, 22, 30, 31, 40, 41, 49, 50, 58, 59, 68, 69, 76, 77, 84, 87, 93, 94, 101, 109, 110, 119, 120, 129, 130, 139, 140, 149, 150
- allUnits** (globale Methode), 8, 11, 14, 18–20, 29, 30, 39, 40, 48, 49, 57, 58, 66, 67
- and** (log. Operator), 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 94, 95, 102, 111, 121, 130, 131, 141, 151
- Arbeit**, 1, 4, 8, 10, 135, 137, 138, 141, 142, 144, 145, 148, 156
- au** (globale Methode), 8, 11, 14, 18, 29, 39, 48, 57, 66
- B** (Sub-Klasse von **G**), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 115–123, 128, 131, 141, 152, 155, 156
- beispiel** (B-Methode), 116, 117, 122
- beispiel** (F1-Methode), 126, 127, 132
- beispiel** (G-Methode), 82, 88
- beispiel** (L1-Methode), 17, 18, 23
- beispiel** (L2-Methode), 27, 28, 33
- beispiel** (L3-Methode), 37, 38, 43
- beispiel** (M-Methode), 64, 65, 70
- beispiel** (N-Methode), 92, 95
- beispiel** (P-Methode), 146, 147, 152
- beispiel** (T1-Methode), 46, 47, 52
- beispiel** (T2-Methode), 55, 56, 61
- beispiel** (TT-Methode), 74, 75, 79
- beispiel** (U-Methode), 98, 99, 103
- beispiel** (V-Methode), 106, 107, 112
- beispiel** (W-Methode), 136, 137, 142
- beispiel** (globale Methode), 8, 11
- Beschleunigung**, 1, 4, 8, 10, 14, 115, 117, 118, 121, 122, 124, 125, 135, 156

- ci (B-Methode), 117
- ci (F1-Methode), 127
- ci (G-Methode), 82
- ci (L1-Methode), 18
- ci (L2-Methode), 28
- ci (L3-Methode), 38
- ci (M-Methode), 65
- ci (N-Methode), 92
- ci (P-Methode), 147
- ci (T1-Methode), 47
- ci (T2-Methode), 56
- ci (TT-Methode), 75
- ci (U-Methode), 99
- ci (V-Methode), 107
- ci (W-Methode), 137
- classInfo (B-Methode), 115–117, 122
- classInfo (F1-Methode), 125–127, 132
- classInfo (G-Methode), 81, 82, 89
- classInfo (L1-Methode), 16–18, 24
- classInfo (L2-Methode), 26–28, 33
- classInfo (L3-Methode), 36–38, 43
- classInfo (M-Methode), 63–65, 71
- classInfo (N-Methode), 91–93, 96
- classInfo (P-Methode), 145–147, 152, 153
- classInfo (T1-Methode), 45–47, 52
- classInfo (T2-Methode), 54–56, 61
- classInfo (TT-Methode), 73–75, 79
- classInfo (U-Methode), 97–99, 103
- classInfo (V-Methode), 105–107, 112, 113
- classInfo (W-Methode), 135–137, 142
- clear (globale Methode), 8, 12, 13
- date (globale Eigenschaft), 2, 15
- description (B-Methode), 116, 117
- description (F1-Methode), 126, 127
- description (G-Methode), 82
- description (L1-Methode), 17, 18
- description (L2-Methode), 27, 28
- description (L3-Methode), 37, 38
- description (M-Methode), 64, 65
- description (N-Methode), 92
- description (P-Methode), 146, 147
- description (T1-Methode), 46, 47
- description (T2-Methode), 55, 56
- description (TT-Methode), 74, 75
- description (U-Methode), 98, 99
- description (V-Methode), 106, 107
- description (W-Methode), 136, 137
- dok (globale Methode), 8, 9, 12, 15–18, 20, 23, 26, 28, 30, 33, 36, 38, 40, 43, 45, 47–49, 52, 54, 56, 58, 61, 63, 65–67, 70, 73, 75, 76, 79, 81–84, 88, 91–93, 95, 97, 99, 100, 103, 105, 107, 108, 112, 115–118, 122, 125–128, 132, 135–138, 142, 145, 147, 148, 152, 157
- Download, 2
- Druck, 8, 13, 157
- druck (globale Methode), 8, 13, 157
- Eigenschaft, 23, 24, 33, 43, 52, 61, 71, 79, 89, 96, 97, 103, 105, 112, 115, 122, 125, 132, 135, 142, 145, 152
- B
 - internal, 122, 123
 - links, 122, 123
 - name, 122, 123
 - oben, 122, 123
 - operator, 122, 123
 - rechts, 122, 123
 - unten, 122, 123
 - v, 122
- F1
 - internal, 132, 133
 - links, 132, 133
 - name, 132, 133
 - oben, 132, 133
 - operator, 132, 133
 - rechts, 132, 133
 - unten, 132, 133
 - v, 132, 133
- G
 - internal, 89, 90
 - links, 89, 90
 - name, 89, 90
 - oben, 89, 90
 - operator, 89, 90
 - rechts, 89, 90
 - unten, 89, 90
 - v, 89
- global
 - date, 2, 15
 - eps, 12, 15
 - modul, 2, 15
 - rndg, 12, 15

- science, 12, 15
- trennz, 12, 15
- version, 2, 15
- L1
 - internal, 24
 - name, 24
 - u, 24
 - v, 24
- L2
 - internal, 33, 34
 - name, 33, 34
 - u, 33, 34
 - v, 33, 34
- L3
 - internal, 43, 44
 - name, 43, 44
 - u, 43, 44
 - v, 43, 44
- M
 - internal, 71
 - name, 71
 - u, 71
 - v, 71
- N
 - internal, 96
 - name, 96
 - v, 96
- P
 - internal, 153
 - links, 153
 - name, 153
 - oben, 153
 - operator, 153
 - rechts, 153
 - unten, 153
 - v, 153
- T1
 - internal, 52
 - name, 52
 - u, 52
 - v, 52
- T2
 - internal, 61, 62
 - name, 61, 62
 - u, 61, 62
 - v, 61, 62
- TT
 - internal, 79
 - name, 79
 - u, 79
 - v, 79
- U
 - internal, 103, 104
 - name, 103, 104
 - u, 104
 - v, 103, 104
 - v1, 103, 104
 - v2, 103, 104
 - v3, 103, 104
- V
 - internal, 113
 - links, 113
 - name, 113
 - oben, 113
 - operator, 113
 - rechts, 113
 - unten, 113
 - v, 113
- W
 - internal, 142, 143
 - links, 142, 143
 - name, 142, 143
 - oben, 142, 143
 - operator, 142, 143
 - rechts, 142, 143
 - unten, 142, 143
 - v, 142, 143
- eps (globale Eigenschaft), 12, 15
- F1 (Sub-Klasse von G), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 125–128, 130–133, 138, 141, 152, 155, 156
- Fehlermeldung, 20, 22, 23, 25, 30, 32, 33, 35, 40, 42–44, 49, 51–53, 58, 60–62, 67, 69, 70, 72, 76, 78–80, 84, 87, 88, 90, 93–95, 100, 102–104, 108, 111, 112, 114, 118, 121, 122, 124, 128, 130, 131, 134, 138, 141, 142, 144, 148, 151, 152, 154
- Flächenmaß, 1, 13, 26, 29, 30, 35
 - amerikanisch, 1, 29, 35
- Frequenz, 8, 13, 157
- frequenz (globale Methode), 8, 13, 157

- G (Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 81–84, 87–90, 95, 102, 105, 107, 108, 110–113, 115–123, 125–128, 130–133, 135–138, 140–143, 145, 147–153, 155, 156
- B (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 115–123, 128, 131, 141, 152, 155, 156
- F1 (Sub-Klasse), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 125–128, 130–133, 138, 141, 152, 155, 156
- P (Sub-Klasse), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 141, 145, 147–153, 155, 156
- V (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 105, 107, 108, 110–113, 121, 131, 141, 152, 155, 156
- W (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 135–138, 140–143, 152, 155, 156
- gcn (globale Methode), 8, 12, 14
- Geschwindigkeit, 1, 4, 8, 10, 13, 14, 84, 105, 107, 108, 111, 114, 157
- getClassNames (globale Methode), 8, 12–14
- getFunctionNames (globale Methode), 8, 12–14
- Gewichtsmaß, 1, 8, 13, 63, 67, 72, 125, 135
 - amerikanisch, 1, 8, 67, 72
 - nicht-metrisch, 67
- gfn (globale Methode), 8, 12, 14
- gi (globale Methode), 8, 12, 14
- globalInfo (globale Methode), 3, 8, 9, 12, 14, 15
- gproc3 (globale Methode), 8, 13, 157
- IDLE, 2
- Import, 2
- Impuls, 8, 13, 84, 157
- impuls (globale Methode), 8, 13, 157
- info (B-Methode), 116–119, 122, 123
- info (F1-Methode), 126–129, 132, 133
- info (G-Methode), 82–84, 89, 90
- info (L1-Methode), 17, 18, 20, 21, 23, 24
- info (L2-Methode), 27, 28, 30, 33, 34
- info (L3-Methode), 37, 38, 40, 43, 44
- info (M-Methode), 64–68, 71
- info (N-Methode), 92, 93, 96
- info (P-Methode), 146–149, 152, 153
- info (T1-Methode), 46–49, 52
- info (T2-Methode), 55, 56, 58, 61, 62
- info (TT-Methode), 74–76, 79
- info (U-Methode), 98–101, 103, 104
- info (V-Methode), 106–109, 112, 113
- info (W-Methode), 136–139, 142, 143
- internal (B-Eigenschaft), 122, 123
- internal (F1-Eigenschaft), 132, 133
- internal (G-Eigenschaft), 89, 90
- internal (L1-Eigenschaft), 24
- internal (L2-Eigenschaft), 33, 34
- internal (L3-Eigenschaft), 43, 44
- internal (M-Eigenschaft), 71
- internal (N-Eigenschaft), 96
- internal (P-Eigenschaft), 153
- internal (T1-Eigenschaft), 52
- internal (T2-Eigenschaft), 61, 62
- internal (TT-Eigenschaft), 79
- internal (U-Eigenschaft), 103, 104
- internal (V-Eigenschaft), 113
- internal (W-Eigenschaft), 142, 143
- joule (globale Methode), 4, 8, 10, 138, 141, 142, 144, 148
- kelvin (globale Methode), 8, 10, 76, 80
- kilogramm (globale Methode), 8–10, 67, 72
- Klasse, 1, 2, 10, 11, 14, 16–24, 26, 28–34, 36, 38–45, 47–54, 56–63, 65–71, 73, 75–84, 87–97, 99–105, 107, 108, 110–113, 115–123, 125–128, 130–133, 135–138, 140–143, 145, 147–153, 155, 156
- G, 1, 11, 23, 32, 42, 51, 60, 70, 78, 81–84, 87–90, 95, 102, 105, 112, 115, 121, 125, 131, 135, 141, 145, 152, 155, 156
- B (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 115–123, 128, 131, 141, 152, 155, 156
- F1 (Sub-Klasse), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 125–128, 130–133, 138, 141, 152, 155, 156

- P (Sub-Klasse), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 141, 145, 147–153, 155, 156
- V (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 105, 107, 108, 110–113, 121, 131, 141, 152, 155, 156
- W (Sub-Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 135–138, 140–143, 152, 155, 156
- L, 14
- L1, 1, 10, 11, 14, 16–24, 32, 42, 51, 60, 70, 78, 84, 88, 95, 102, 108, 112, 117, 118, 121, 127, 131, 137, 138, 141, 152, 155, 156
- L2, 1, 10, 11, 23, 26, 28–34, 42, 51, 60, 70, 78, 84, 88, 95, 102, 112, 121, 131, 141, 148, 152, 155, 156
- L3, 1, 10, 11, 23, 32, 36, 38–44, 51, 60, 70, 78, 84, 88, 95, 102, 112, 121, 131, 141, 152, 155, 156
- M, 1, 10, 11, 23, 32, 42, 51, 60, 63, 65–71, 78, 84, 88, 95, 102, 112, 121, 127, 128, 131, 137, 141, 152, 155, 156
- N, 2, 11, 23, 32, 42, 51, 60, 70, 78, 84, 88, 91–96, 102, 112, 121, 131, 141, 152, 155, 156
- T, 14
- T1, 1, 11, 14, 23, 32, 42, 45, 47–53, 60, 70, 78, 84, 88, 95, 97, 100, 102, 108, 112, 121, 131, 141, 148, 152, 155, 156
- U (Sub-Klasse), 2, 11, 47, 97, 99–104, 155, 156
- T2, 1, 11, 23, 32, 42, 51, 54, 56–62, 70, 78, 84, 88, 95, 102, 112, 117, 118, 121, 127, 131, 137, 141, 152, 155, 156
- TT, 1, 10, 11, 23, 32, 42, 51, 60, 70, 73, 75–80, 84, 88, 95, 102, 112, 121, 131, 141, 152, 155, 156
- Konvertierung, 2, 5, 6, 24, 25, 34, 35, 44, 53, 62, 71, 72, 80, 90, 104, 113, 114, 123, 124, 133, 134, 143, 144, 153, 154
- Kraft, 1, 4, 8, 11, 13, 125, 127, 128, 130, 131, 134, 135, 138, 145, 156
- L (Klasse), 14
- L1 (Klasse), 1, 10, 11, 14, 16–24, 32, 42, 51, 60, 70, 78, 84, 88, 95, 102, 108, 112, 117, 118, 121, 127, 131, 137, 138, 141, 152, 155, 156
- L2 (Klasse), 1, 10, 11, 23, 26, 28–34, 42, 51, 60, 70, 78, 84, 88, 95, 102, 112, 121, 131, 141, 148, 152, 155, 156
- L3 (Klasse), 1, 10, 11, 23, 32, 36, 38–44, 51, 60, 70, 78, 84, 88, 95, 102, 112, 121, 131, 141, 152, 155, 156
- Längenmaß, 1, 8, 13, 16, 19, 20, 25, 105, 108, 115, 118, 135, 138, 145
 - amerikanisch, 1, 8, 19, 20
 - Druckergewerbe, 1, 19
- Leistung, 1, 4, 8, 11, 13, 145, 147, 148, 151, 152, 154
- links (B-Eigenschaft), 122, 123
- links (F1-Eigenschaft), 132, 133
- links (G-Eigenschaft), 89, 90
- links (P-Eigenschaft), 153
- links (V-Eigenschaft), 113
- links (W-Eigenschaft), 142, 143
- literpro100km (globale Methode), 8, 13, 157
- literprom2 (globale Methode), 8, 13, 157
- log. Operator
 - and, 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 94, 95, 102, 111, 121, 130, 131, 141, 151
 - not, 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 88, 94, 95, 102, 111, 121, 130, 131, 141, 151
 - or, 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 88, 94, 95, 102, 111, 121, 130, 131, 141, 151
- longweightsToM (globale Methode), 8, 67
- M (Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 63, 65–71, 78, 84, 88, 95, 102, 112, 121, 127, 128, 131, 137, 141, 152, 155, 156
- Maßeinheit, 1–3, 18–20, 24, 25, 29, 30, 34, 35, 39, 40, 44, 48, 49, 53, 57, 58, 62, 66, 67, 71, 72, 76, 80, 83, 90, 93, 100, 104, 108, 114, 117, 124, 127, 134, 137, 144, 148, 154
- meter (globale Methode), 8, 10, 19, 20, 25

- meter2 (globale Methode), 8, 10, 29, 30, 35
- meter3 (globale Methode), 8, 10, 39, 40, 44
- meterS (globale Methode), 4, 8, 10, 14, 84, 108, 111, 114
- meterS2 (globale Methode), 4, 8, 10, 14, 118, 121, 122, 124
- Methode
 - B
 - beispiel, 116, 117, 122
 - ci, 117
 - classInfo, 115–117, 122
 - description, 116, 117
 - info, 116–119, 122, 123
 - to, 116, 117, 123, 124
 - F1
 - beispiel, 126, 127, 132
 - ci, 127
 - classInfo, 125–127, 132
 - description, 126, 127
 - info, 126–129, 132, 133
 - to, 126, 127, 133, 134
 - G
 - beispiel, 82, 88
 - ci, 82
 - classInfo, 81, 82, 89
 - description, 82
 - info, 82–84, 89, 90
 - to, 82, 83, 90
 - global
 - alle, 8, 11, 13, 21, 22, 30, 31, 40, 41, 49, 50, 58, 59, 68, 69, 76, 77, 84, 87, 93, 94, 101, 109, 110, 119, 120, 129, 130, 139, 140, 149, 150
 - allUnits, 8, 11, 14, 18–20, 29, 30, 39, 40, 48, 49, 57, 58, 66, 67
 - au, 8, 11, 14, 18, 29, 39, 48, 57, 66
 - beispiel, 8, 11
 - clear, 8, 12, 13
 - dok, 8, 9, 12, 15–18, 20, 23, 26, 28, 30, 33, 36, 38, 40, 43, 45, 47–49, 52, 54, 56, 58, 61, 63, 65–67, 70, 73, 75, 76, 79, 81–84, 88, 91–93, 95, 97, 99, 100, 103, 105, 107, 108, 112, 115–118, 122, 125–128, 132, 135–138, 142, 145, 147, 148, 152, 157
 - druck, 8, 13, 157
 - frequenz, 8, 13, 157
 - gcn, 8, 12, 14
 - getClassNames, 8, 12–14
 - getFunctionNames, 8, 12–14
 - gfn, 8, 12, 14
 - gi, 8, 12, 14
 - globalInfo, 3, 8, 9, 12, 14, 15
 - gproc3, 8, 13, 157
 - impuls, 8, 13, 157
 - joule, 4, 8, 10, 138, 141, 142, 144, 148
 - kelvin, 8, 10, 76, 80
 - kilogramm, 8–10, 67, 72
 - literpro100km, 8, 13, 157
 - literprom2, 8, 13, 157
 - longweightsToM, 8, 67
 - meter, 8, 10, 19, 20, 25
 - meter2, 8, 10, 29, 30, 35
 - meter3, 8, 10, 39, 40, 44
 - meterS, 4, 8, 10, 14, 84, 108, 111, 114
 - meterS2, 4, 8, 10, 14, 118, 121, 122, 124
 - mgprodl, 8, 13, 157
 - mpros, 8, 13, 157
 - mS, 8, 10, 14
 - mS2, 8, 10, 14
 - myfiToL1, 8, 19, 20
 - newton, 4, 8, 11, 128, 130, 131, 134, 138
 - PS, 11, 148, 154
 - sekunde, 8, 11, 48, 49, 53
 - sekunde2, 8, 11, 57, 62
 - setVar, 8, 12, 14, 15
 - sign, 8, 12, 13
 - sv, 8, 12, 14
 - typ, 8, 12, 13
 - Upromin, 8, 13, 157
 - watt, 4, 8, 11, 148, 151, 152, 154
 - ymdToT1, 8, 49
 - L1
 - beispiel, 17, 18, 23
 - ci, 18
 - classInfo, 16–18, 24
 - description, 17, 18
 - info, 17, 18, 20, 21, 23, 24
 - to, 6, 17, 18, 24, 25
 - L2
 - beispiel, 27, 28, 33
 - ci, 28

- classInfo, 26–28, 33
- description, 27, 28
- info, 27, 28, 30, 33, 34
- to, 6, 27, 28, 34, 35
- L3
 - beispiel, 37, 38, 43
 - ci, 38
 - classInfo, 36–38, 43
 - description, 37, 38
 - info, 37, 38, 40, 43, 44
 - to, 37, 38, 44
- M
 - beispiel, 64, 65, 70
 - ci, 65
 - classInfo, 63–65, 71
 - description, 64, 65
 - info, 64–68, 71
 - to, 6, 64–66, 71, 72
 - toLongweights, 64–66, 71, 72
- N
 - beispiel, 92, 95
 - ci, 92
 - classInfo, 91–93, 96
 - description, 92
 - info, 92, 93, 96
- P
 - beispiel, 146, 147, 152
 - ci, 147
 - classInfo, 145–147, 152, 153
 - description, 146, 147
 - info, 146–149, 152, 153
 - to, 146, 147, 153, 154
- T1
 - beispiel, 46, 47, 52
 - ci, 47
 - classInfo, 45–47, 52
 - description, 46, 47
 - info, 46–49, 52
 - to, 6, 46–48, 53
 - toU, 46–48
 - toYMD, 46–48, 53
- T2
 - beispiel, 55, 56, 61
 - ci, 56
 - classInfo, 54–56, 61
 - description, 55, 56
 - info, 55, 56, 58, 61, 62
 - to, 55, 56, 62
- TT
 - beispiel, 74, 75, 79
 - ci, 75
 - classInfo, 73–75, 79
 - description, 74, 75
 - info, 74–76, 79
 - to, 6, 74, 75, 80
- U
 - beispiel, 98, 99, 103
 - ci, 99
 - classInfo, 97–99, 103
 - description, 98, 99
 - info, 98–101, 103, 104
 - to, 98–100, 104
 - toYMD, 98–100, 104
- V
 - beispiel, 106, 107, 112
 - ci, 107
 - classInfo, 105–107, 112, 113
 - description, 106, 107
 - info, 106–109, 112, 113
 - to, 106, 107, 113, 114
- W
 - beispiel, 136, 137, 142
 - ci, 137
 - classInfo, 135–137, 142
 - description, 136, 137
 - info, 136–139, 142, 143
 - to, 137, 143, 144
- mgprodl (globale Methode), 8, 13, 157
- modul (globale Eigenschaft), 2, 15
- mpros (globale Methode), 8, 13, 157
- mS (globale Methode), 8, 10, 14
- mS2 (globale Methode), 8, 10, 14
- myfiToL1 (globale Methode), 8, 19, 20
- N (Klasse), 2, 11, 23, 32, 42, 51, 60, 70, 78, 84, 88, 91–96, 102, 112, 121, 131, 141, 152, 155, 156
- name (B-Eigenschaft), 122, 123
- name (F1-Eigenschaft), 132, 133
- name (G-Eigenschaft), 89, 90
- name (L1-Eigenschaft), 24
- name (L2-Eigenschaft), 33, 34
- name (L3-Eigenschaft), 43, 44
- name (M-Eigenschaft), 71

- name (N-Eigenschaft), 96
- name (P-Eigenschaft), 153
- name (T1-Eigenschaft), 52
- name (T2-Eigenschaft), 61, 62
- name (TT-Eigenschaft), 79
- name (U-Eigenschaft), 103, 104
- name (V-Eigenschaft), 113
- name (W-Eigenschaft), 142, 143
- newton (globale Methode), 4, 8, 11, 128, 130, 131, 134, 138
- not (log. Operator), 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 88, 94, 95, 102, 111, 121, 130, 131, 141, 151
- oben (B-Eigenschaft), 122, 123
- oben (F1-Eigenschaft), 132, 133
- oben (G-Eigenschaft), 89, 90
- oben (P-Eigenschaft), 153
- oben (V-Eigenschaft), 113
- oben (W-Eigenschaft), 142, 143
- Operator
 - ****, 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 111, 112, 121, 122, 131, 141, 142, 151, 152
 - ***, 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 111, 112, 121, 122, 131, 141, 142, 151, 152, 155
 - +**, 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 79, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152
 - , 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 79, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152
 - /**, 5, 23, 32, 33, 42, 43, 51, 52, 60, 61, 70, 78, 88, 95, 102, 103, 111, 112, 121, 122, 131, 141, 142, 151, 152, 155, 156
- operator (B-Eigenschaft), 122, 123
- operator (F1-Eigenschaft), 132, 133
- operator (G-Eigenschaft), 89, 90
- operator (P-Eigenschaft), 153
- operator (V-Eigenschaft), 113
- operator (W-Eigenschaft), 142, 143
- or (log. Operator), 5, 22, 23, 32, 42, 51, 60, 69, 70, 78, 87, 88, 94, 95, 102, 111, 121, 130, 131, 141, 151
- P (Sub-Klasse von G), 1, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 141, 145, 147–153, 155, 156
- PS (globale Methode), 11, 148, 154
- Python-Benutzerfläche, 2
- Python-Benutzeroberfläche, 2
- Python-Pfad, 2
- PYTHONPATH, 2
- rechts (B-Eigenschaft), 122, 123
- rechts (F1-Eigenschaft), 132, 133
- rechts (G-Eigenschaft), 89, 90
- rechts (P-Eigenschaft), 153
- rechts (V-Eigenschaft), 113
- rechts (W-Eigenschaft), 142, 143
- rndg (globale Eigenschaft), 12, 15
- science (globale Eigenschaft), 12, 15
- sekunde (globale Methode), 8, 11, 48, 49, 53
- sekunde2 (globale Methode), 8, 11, 57, 62
- setVar (globale Methode), 8, 12, 14, 15
- sign (globale Methode), 8, 12, 13
- sv (globale Methode), 8, 12, 14
- T (Klasse), 14
- T1 (Klasse), 1, 2, 11, 14, 23, 32, 42, 45, 47–53, 60, 70, 78, 84, 88, 95, 97, 99–104, 108, 112, 121, 131, 141, 148, 152, 155, 156
- U (Sub-Klasse), 2, 11, 47, 97, 99–104, 155, 156
- T2 (Klasse), 1, 11, 23, 32, 42, 51, 54, 56–62, 70, 78, 84, 88, 95, 102, 112, 117, 118, 121, 127, 131, 137, 141, 152, 155, 156
- Temperaturmaß, 1, 73, 80
- to (B-Methode), 116, 117, 123, 124
- to (F1-Methode), 126, 127, 133, 134
- to (G-Methode), 82, 83, 90
- to (L1-Methode), 6, 17, 18, 24, 25
- to (L2-Methode), 6, 27, 28, 34, 35
- to (L3-Methode), 37, 38, 44
- to (M-Methode), 6, 64–66, 71, 72
- to (P-Methode), 146, 147, 153, 154
- to (T1-Methode), 6, 46–48, 53
- to (T2-Methode), 55, 56, 62
- to (TT-Methode), 6, 74, 75, 80
- to (U-Methode), 98–100, 104

- to (V-Methode), 106, 107, 113, 114
- to (W-Methode), 137, 143, 144
- toLongweights (M-Methode), 64–66, 71, 72
- toU (T1-Methode), 46–48
- toYMD (T1-Methode), 46–48, 53
- toYMD (U-Methode), 98–100, 104
- trennz (globale Eigenschaft), 12, 15
- TT (Klasse), 1, 10, 11, 23, 32, 42, 51, 60, 70, 73, 75–80, 84, 88, 95, 102, 112, 121, 131, 141, 152, 155, 156
- typ (globale Methode), 8, 12, 13
- U (Sub-Klasse von T1), 2, 11, 47, 97, 99–104, 155, 156
- u (L1-Eigenschaft), 24
- u (L2-Eigenschaft), 33, 34
- u (L3-Eigenschaft), 43, 44
- u (M-Eigenschaft), 71
- u (T1-Eigenschaft), 52
- u (T2-Eigenschaft), 61, 62
- u (TT-Eigenschaft), 79
- u (U-Eigenschaft), 104
- Uhrzeit, 2, 97, 104
- unten (B-Eigenschaft), 122, 123
- unten (F1-Eigenschaft), 132, 133
- unten (G-Eigenschaft), 89, 90
- unten (P-Eigenschaft), 153
- unten (V-Eigenschaft), 113
- unten (W-Eigenschaft), 142, 143
- Upromin (globale Methode), 8, 13, 157
- V (Sub-Klasse von G), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 105, 107, 108, 110–113, 121, 131, 141, 152, 155, 156
- v (B-Eigenschaft), 122
- v (F1-Eigenschaft), 132, 133
- v (G-Eigenschaft), 89
- v (L1-Eigenschaft), 24
- v (L2-Eigenschaft), 33, 34
- v (L3-Eigenschaft), 43, 44
- v (M-Eigenschaft), 71
- v (N-Eigenschaft), 96
- v (P-Eigenschaft), 153
- v (T1-Eigenschaft), 52
- v (T2-Eigenschaft), 61, 62
- v (TT-Eigenschaft), 79
- v (U-Eigenschaft), 103, 104
- v (V-Eigenschaft), 113
- v (W-Eigenschaft), 142, 143
- v1 (U-Eigenschaft), 103, 104
- v2 (U-Eigenschaft), 103, 104
- v3 (U-Eigenschaft), 103, 104
- Vergleich
 - <=, 5, 22, 23, 31, 32, 41, 42, 50, 51, 59, 60, 69, 70, 77, 78, 87, 94, 95, 101, 102, 111, 120, 121, 130, 131, 140, 141, 150, 151
 - <, 5, 22, 31, 41, 50, 59, 69, 77, 87, 94, 101, 110, 120, 130, 140, 150
 - ==, 5, 22, 23, 31, 32, 41, 42, 50, 51, 59, 60, 69, 70, 77, 78, 87, 88, 94, 95, 101, 102, 111, 120, 121, 130, 131, 140, 141, 151
 - >=, 5, 22, 31, 32, 41, 42, 50, 51, 59, 60, 69, 77, 87, 94, 101, 111, 120, 121, 130, 140, 141, 151
 - >, 5, 22, 31, 41, 50, 59, 69, 77, 87, 94, 101, 111, 120, 130, 140, 150
 - !=, 22, 31, 32, 41, 42, 50, 51, 59, 60, 69, 77, 78, 87, 94, 101, 102, 111, 120, 121, 130, 140, 141, 151
- version (globale Eigenschaft), 2, 15
- Volumenmaß, 1, 13, 36, 39, 40, 44
 - amerikanisch, 1, 39, 44
- Vorzeichen
 - +, 19, 20, 29, 30, 39, 40, 48, 57, 67, 83, 84, 93, 100, 108, 118, 128, 138, 148
 - , 19, 20, 29, 30, 39, 40, 48, 49, 57, 67, 83, 84, 93, 100, 108, 118, 128, 138, 148
- W (Sub-Klasse von G), 1, 10, 11, 23, 32, 42, 51, 60, 70, 78, 88, 95, 102, 112, 121, 131, 135–138, 140–143, 152, 155, 156
- watt (globale Methode), 4, 8, 11, 148, 151, 152, 154
- Windows, 2
- ymdToT1 (globale Methode), 8, 49
- Zeitmaß, 1, 13, 45, 49, 53, 54, 100, 104, 108, 115, 145