

# chemmacros v1.1

2011/06/22

Clemens NIEDERBERGER

<http://www.mychemistry.eu/>  
[contact@mychemistry.eu](mailto:contact@mychemistry.eu)

‘chemmacros’ ist eine Sammlung von Hilfs-Makros für Chemiker. Sie sind dazu gedacht, das Schreiben von Chemie-Dokumenten mit  $\text{\LaTeX 2}_{\epsilon}$  schneller und bequemer zu machen. Das betrifft unter anderem Nomenklatur, Oxidationszahlen, thermodynamische Daten, Newman-Projektionen, usw.

## Inhaltsverzeichnis

<b>1</b>	<b>Lizenz, Voraussetzungen</b>	<b>3</b>
<b>2</b>	<b>Paket-Optionen</b>	<b>4</b>
<b>3</b>	<b>Teilchen, Ionen und ein Symbol</b>	<b>4</b>
<b>4</b>	<b>Stereo-Deskriptoren, Nomenklatur, lateinische Ausdrücke</b>	<b>5</b>
4.1	Stereo-Deskriptoren und Nomenklatur . . . . .	5
4.2	Lateinische Ausdrücke . . . . .	7
<b>5</b>	<b>Einheiten für ‘siunitx’</b>	<b>8</b>
<b>6</b>	<b>Säure/Base</b>	<b>9</b>
<b>7</b>	<b>Oxidationszahlen und (echte) Ladungen</b>	<b>9</b>
7.1	Ionenladungen . . . . .	9
7.2	Oxidationszahlen . . . . .	9
7.3	Partialladungen und ähnliches . . . . .	10
<b>8</b>	<b>Reaktionsmechanismen</b>	<b>11</b>
<b>9</b>	<b>Redoxreaktionen</b>	<b>12</b>
<b>10</b>	<b>(Standard) Zustand, Thermodynamik</b>	<b>14</b>
10.1	Thermodynamische Größen . . . . .	14
10.1.1	Neue Größen definieren . . . . .	15
10.1.2	Größen umdefinieren . . . . .	16
10.2	State . . . . .	16
<b>11</b>	<b>Spektroskopie</b>	<b>17</b>
<b>12</b>	<b>Befehle für ‘mhchem’</b>	<b>19</b>
12.1	Reaktionsumgebungen . . . . .	20
12.1.1	Durch ‘chemmacros’ definiert . . . . .	20
12.1.2	Eigene Reaktionen . . . . .	22
12.2	Phasen . . . . .	23
12.3	Beschriftungen . . . . .	24
<b>13</b>	<b>Newman-Projektionen</b>	<b>26</b>
<b>14</b>	<b>p-Orbitale</b>	<b>27</b>
<b>15</b>	<b>Liste der Befehle</b>	<b>30</b>

# 1 Lizenz, Voraussetzungen

‘chemmacros’ v1.1 steht unter der L<sup>A</sup>T<sub>E</sub>X Project Public License Version 1.3 oder später. (<http://www.latex-project.org/lppl.txt>)

‘chemmacros’ ruft intern die Pakete ‘amsmath’<sup>1)</sup>, ‘ifthen’<sup>2)</sup>, ‘siunitx’<sup>3)</sup>, ‘xparse’<sup>4)</sup> und ‘tikz’ (TikZ = ‘pgf’<sup>5)</sup>) sowie die TikZlibrary `calc` auf. Sind sie nicht vorhanden, wird eine Fehlermeldung ausgegeben.

‘siunitx’ benötigt L<sup>A</sup>T<sub>E</sub>X3-Unterstützung, die durch die Bundles ‘expl3’<sup>6)</sup> und ‘xpackages’<sup>7)</sup> bereitgestellt wird. ‘xparse’ ist Teil des ‘xpackages’-Bundles. In der Folge benötigt auch ‘chemmacros’ L<sup>A</sup>T<sub>E</sub>X3-Unterstützung.

Manche Befehle haben unterschiedliche Definitionen, je nachdem ob andere Pakete vom Benutzer *vorher* geladen wurden oder nicht. Manche Befehle werden auch nur definiert, wenn ein bestimmtes Paket geladen ist. Das betrifft die Pakete ‘bpchem’<sup>8)</sup>, ‘chemstyle’<sup>9)</sup> und ‘mhchem’<sup>10)</sup>. An entsprechender Stelle in der Dokumentation wird auf die Unterschiede konkret hingewiesen.

Für die Paketoption `bpchem` (Abschnitt 2) wird das Paket ‘bpchem’ benötigt.

Für die Paketoption `xspace` (Abschnitt 2) wird das Paket ‘xspace’<sup>11)</sup> benötigt.

Wenn der Benutzer das Paket ‘mhchem’ lädt, so werden außerdem die Pakete ‘mathtools’<sup>12)</sup> und ‘environ’<sup>13)</sup> benötigt.

**Bitte beachten Sie, dass sich mit der Version 1.1 die Paketoptionen verändert haben.**

- 
- 1) <http://www.ctan.org/pkg/amsmath>
  - 2) <http://www.ctan.org/pkg/ifthen>
  - 3) <http://www.ctan.org/pkg/siunitx>
  - 4) <http://www.ctan.org/pkg/xparse>
  - 5) <http://www.ctan.org/pkg/pgf>
  - 6) <http://www.ctan.org/pkg/expl3>
  - 7) <http://www.ctan.org/pkg/xpackages>
  - 8) <http://www.ctan.org/pkg/bpchem>
  - 9) <http://www.ctan.org/pkg/chemstyle>
  - 10) <http://www.ctan.org/pkg/mhchem>
  - 11) <http://www.ctan.org/pkg/xspace>
  - 12) <http://www.ctan.org/pkg/mathtools>
  - 13) <http://www.ctan.org/pkg/environ>

## 2 Paket-Optionen

NEU v.1.1

‘chemmacros’ hat vier Paketoptionen:

**bpchem** Durch diese Option wird zum einen das Paket ‘bpchem’ geladen und zum anderen das Aussehen des `\NMR`-Befehls dem der ‘bpchem’-Befehle `\HNMR` und `\CNMR` angepasst. Ohne Option:  $^1\text{H-NMR}$ :  $\delta$ ; Mit Option:  $^1\text{H-NMR}$ :  $\delta$ ;

**circled** Manche Chemiker – darunter ich – bevorzugen umkreiste Ladungssymbole, um sie von dem Additions- und Subtraktionssymbol klar abzugrenzen. Per Default werden sie bei ‘chemmacros’ allerdings *nicht* umkreist angezeigt (+ und –). Mit der Option **circled** verwenden alle Befehle von ‘chemmacros’ die Symbole *mit* Kreis ( $\oplus$  und  $\ominus$ ).

**german** Die Option ändert den Befehl `\pKa` von  $pK_A$  in  $pK_S$  und die Befehle `\solid` und `\liquid` von  $_{(s)}$  und  $_{(l)}$  in  $_{(f)}$  bzw.  $_{(fl)}$ .

**xspace** Mit der Option bekommen folgende Befehle ein `\xspace` angehängt: `\Hp1` `\HtO` `\water` `\Hyd` `\HtO` `\pH` `\pOH` `\pKa` `\pKb` `\cis` `\trans` `\insitu` `\abinitio` `\mech` `\NMR`  $\text{H}^\oplus$   $\text{H}_3\text{O}^\oplus$   $\text{H}_2\text{O}$   $\text{OH}^\ominus$   $\text{H}_3\text{O}^\oplus$   $\text{pH}$   $\text{pOH}$   $pK_S$   $pK_B$  *cis* *trans* *in situ* *ab initio*  $\text{S}_\text{N}$   $^1\text{H-NMR}$ :  $\delta$

Bitte beachten Sie, dass die beiden Befehle `\cis` und `\trans` auch durch das ‘bpchem’-Paket definiert werden. Falls Sie das Paket laden, werden sie durch ‘chemmacros’ umdefiniert. Bei ‘bpchem’ haben sie *immer* ein angehängtes `\xspace`, mit ‘chemmacros’ nur mit Option **xspace**. Abgesehen davon sind sie identisch.

## 3 Teilchen, Ionen und ein Symbol

Eine Reihe einfacher Makros, um häufig gebrauchte Teilchen sowie ein Symbol darzustellen. Beachten Sie, dass die Darstellung von den verwendeten Paketoptionen abhängt, siehe Abschnitt 2.

- `\Hp1`  $\text{H}^\oplus$  (Proton)
- `\Hyd`  $\text{OH}^\ominus$  (Hydroxid)
- `\HtO`  $\text{H}_3\text{O}^\oplus$  (Oxonium) (**H** three **O**)
- `\water`  $\text{H}_2\text{O}$

NEU v.1.1

NEU v.1.1

- `\el`  $e^{\ominus}$  (Elektron)

- `\prt`  $p^{\oplus}$  (Proton)

NEU v.1.1

- `\ntr`  $n^0$  (Neutron)

NEU v.1.1

- `\Nu`  $\text{Nu}^{\ominus}$  (Nucleophil)

- `\El`  $E^{\oplus}$  (Elektrophil)

- `\transitionstatesymbol` ≠ Übergangszustandssymbol (verwendet ‘*TikZ*’)

Diese Befehle sind sowohl im Text- wie im Mathematikmodus einsetzbar. Je nachdem, ob ‘mhchem’ geladen wurde oder nicht, werden Atome in den `\cf{}`-Befehl von ‘mhchem’ oder in eine `\mbox{}` geschrieben.

NEU v.1.1

Daneben gibt es noch einen weiteren Befehl, mit dem Radikale mit Ladungen und Index geschrieben werden können.

- `\R[<sign>]{<subscript>}` z. B. `\R[+]{tert}` `\R[-]{sek}` `\R{prim}`  $R_{\text{tert}}^{\oplus}$   $R_{\text{sek}}^{\ominus}$   
 $R_{\text{prim}}$

## 4 Stereo-Deskriptoren, Nomenklatur, lateinische Ausdrücke

### 4.1 Stereo-Deskriptoren und Nomenklatur

Die folgenden Makros sollen das Schreiben von IUPAC-Namen etwas erleichtern:

- Cahn-Ingold-Prelog:

– `\Rcip` ( $R$ )

– `\Scip` ( $S$ )

– `\cip{<conf>}` z. B.: `\cip{R,S}` ( $R,S$ )

- Fischer:

– `\Dfi`  $D$

– `\Lfi`  $L$

- cis/trans und zusammen/entgegen:

– `\Z` (*Z*)

– `\E` (*E*)

– `\cis` *cis* (Dieser Befehl wird auch von ‘bpchem’ definiert. ‘chemmacros’ schreibt in diesem Fall die Definition leicht um, siehe Abschnitt 2.)

– `\trans` *trans* (Dieser Befehl wird auch von ‘bpchem’ definiert. ‘chemmacros’ schreibt in diesem Fall die Definition leicht um, siehe Abschnitt 2.)

- ortho/meta/para:

– `\ortho` *o*

– `\meta` *m*

– `\para` *p*

Anzeigen der absoluten Konfiguration (verwendet ‘**TikZ**’):

• `\Rconf[<letter>]` `\Rconf:`  `\Rconf[]:` 

• `\Sconf[<letter>]` `\Sconf:`  `\Sconf[]:` 

Beispiele:

`1 \Dfi-Weinsäure = \cip{2S,3S}-Weinsäure`

D-Weinsäure = (2*S*,3*S*)-Weinsäure

`1 \Dfi-($-$)-Threose = \cip{2S,3R}-($-$)-2,3,4-Trihydroxybutanal`

D-(–)-Threose = (2*S*,3*R*)-(–)-2,3,4-Trihydroxybutanal

`1 \cis-2-Buten = \Z-2-Butene, \cip{2E,4Z}-Hexadiene`

*cis*-2-Buten = (*Z*)-2-Butene, (2*E*,4*Z*)-Hexadiene

`1 \meta-Xylol = 1,3-Dimethylbenzene`

*m*-Xylol = 1,3-Dimethylbenzene

```

1 % mit 'bpchem's Befehl \IUPAC:
2 \IUPAC{\Dfi\Wein\|säure} = \IUPAC{\cip{2S,3S}\Wein\|sä
   ure}, \IUPAC{\Dfi\-(\$-\$)\Threose} = \IUPAC{\cip{2S,3R
   }\-(\$-\$)\-2,3,4-Tri\|hydroxy\|butanal}

```

D-Weinsäure = (2*S*,3*S*)-Weinsäure, D-(−)-Threose = (2*S*,3*R*)-(−)-2,3,4-Trihydroxybutanal

Im letzten Beispiel wurde der Befehl `\IUPAC` des Pakets 'bpchem' eingesetzt.

Das Aussehen hängt natürlich von gewählten Font ab:

```

1 \cip{2S,3R} \E \Z \Dfi \Lfi \par
2 \fontfamily{ptm}\selectfont
3 \cip{2S,3R} \E \Z \Dfi \Lfi \par
4 \fontfamily{ppl}\selectfont
5 \cip{2S,3R} \E \Z \Dfi \Lfi

```

(2*S*,3*R*) (E)(Z)<sub>DL</sub>

(2*S*,3*R*) (E)(Z)<sub>DL</sub>

(2*S*,3*R*) (E)(Z)<sub>DL</sub>

## 4.2 Lateinische Ausdrücke

NEU v.1.1

Zuletzt gibt es noch zwei Befehle für gebräuchliche lateinische Ausdrücke.

- `\insitu` *in situ*
- `\abinitio` *ab initio*

Falls ebenfalls das Paket 'chemstyle' geladen wurde<sup>1)</sup>, wird für die Definition der 'chemstyle'-Befehl `\latin` verwendet. Das bedeutet, dass das Schriftbild dann durch die 'chemstyle'-Einstellung `abbremph` beeinflusst wird:

```

1 \insitu, \abinitio\
2 \cstsetup{abbremph=false}
3 \insitu, \abinitio

```

---

<sup>1)</sup> 'chemstyle' definiert weitere ähnliche Befehle wie *et al.*, *in vacuo*.

*in situ, ab initio*  
in situ, ab initio

Falls ‘chemstyle’ nicht geladen wurde, werden sie *kursiv* gesetzt.

## 5 Einheiten für ‘siunitx’

NEU v.1.1

Manche nicht-SI-Einheiten sind in der Chemie sehr gebräuchlich. ‘siunitx’ bietet den Befehl `\DeclareSIUnit{<command>}{<unit>}`, mit dem man beliebige Einheiten definieren kann. ‘chemmacros’ stellt einige damit definierte Einheiten zur Verfügung. Wie alle ‘siunitx’-Einheiten sind sie nur innerhalb von `\SI{<num>}{<unit>}` und `\si{<unit>}` definiert.

- `\atmosphere` atm
- `\atm` atm
- `\calory` cal
- `\cal` cal
- `\cmc` cm<sup>3</sup> <sup>1)</sup>
- `\molar` mol dm<sup>-3</sup> <sup>1)</sup>
- `\moLar` mol L<sup>-1</sup>
- `\Molar` M <sup>1)</sup>
- `\MolMass` g mol<sup>-1</sup>
- `\normal` N
- `\torr` torr

Übrigens: `\mmHg` mmHg ist durch ‘siunitx’ und ‘chemstyle’ bereits definiert.

---

<sup>1)</sup> Diese Einheiten werden auch durch ‘chemstyle’ definiert. Sie werden durch ‘chemmacros’ nur definiert, wenn ‘chemstyle’ nicht geladen wird.



## 6 Säure/Base

NEU v.1.1

Einfache Darstellung von  $pH$ ,  $pK_S$  ...

- `\pH`  $pH$
- `\pOH`  $pOH$
- `\pKa[<num>]`  $pK_S$ , `\pKa[1]`  $pK_{S1}$
- `\pKb[<num>]`  $pK_B$ , `\pKb[1]`  $pK_{B1}$

Diese Befehle sind sowohl im Text- wie im Mathematikmodus und im `\ce`-Befehl von ‘mhchem’ einsetzbar. Der Befehl `\pKa` hängt von der Paketoption `german` ab, siehe Abschnitt 2.

## 7 Oxidationszahlen und (echte) Ladungen

### 7.1 Ionenladungen

Einfaches Darstellen von Ladungen:

- `\pch[<number>]` positive Ladung (**plus** + **charge**): `\pch`  $\oplus$ , `Na\pch`  $Na^\oplus$ , `Ca\pch[2]`  $Ca^{2\oplus}$
- `\mch[<number>]` negative Ladung (**minus** + **charge**): `\mch`  $\ominus$ , `F\mch`  $F^\ominus$ , `S\mch[2]`  $S^{2\ominus}$

### 7.2 Oxidationszahlen

Setzen von Oxidationszahlen:

- `\ox{<number>,<atom>}` setzt  $\text{<number>}$  über  $\text{<atom>}$ ; `\ox{+1,Na}`, `\ox{+I,Na}`,  
`\ox{-2,S}`, `\ox{-II,S}`  $\overset{+1}{Na}$ ,  $\overset{+I}{Na}$ ,  $\overset{-2}{S}$ ,  $\overset{-II}{S}$

Wenn das Paket ‘mhchem’ geladen wurde, wird  $\text{<atom>}$  innerhalb des `\cf`-Befehls gesetzt: `\ox{+II,Ca}\ox{-I,F2}`  $\overset{+II}{Ca}\overset{-I}{F_2}$ . Ohne ‘mhchem’ funktioniert das so nicht ( $\overset{+II}{Ca}\overset{-I}{F_2}$ )

und die konventionelle Methode muss verwendet werden:  $\text{CaF}_2$ .

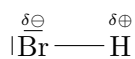
### 7.3 Partialladungen und ähnliches

Vielleicht selten gebraucht, nichtsdestoweniger nützlich:

- `\delp`  $\delta\oplus$  (delta + plus)
- `\delm`  $\delta\ominus$  (delta + minus)

Diese Makros können zum Beispiel mit dem `\ox`-Befehl eingesetzt werden oder zusammen mit dem Paket ‘chemfig’<sup>1)</sup>:

```
1 \ox{\delp,H}\ox{\delm,Cl}\par
2 \chemfig{\chemabove[3pt]{\lewis{246,Br}}{\delm}-\chemabove[3pt]{H}{\delp}}
```

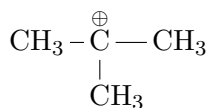


Auch die beiden folgenden Makros sind v.a. beim Einsatz mit ‘chemfig’ nützlich.

- `\scrp`  $\oplus$ (scriptstyle + plus)
- `\scrm`  $\ominus$ (scriptstyle + minus)

Zum Beispiel:

```
1 {\setatomsep{1.8em}\chemfig{CH_3-\chemabove{C}{\scrp}(-[6]C|H_3)-\vphantom{H_3}CH_3}}
```



## 8 Reaktionsmechanismen

NEU v.1.1

Der Befehl

```
1 \mech[<type>]
```

erlaubt die Angabe der verbreitetsten Reaktionsmechanismen. <type> kann einer der folgenden Werte annehmen:

- <type>= (leer, kein opt. Argument) nukleophile Substitution \mech S<sub>N</sub>
- <type>=1 unimolekulare nukleophile Substitution \mech[1] S<sub>N</sub>1
- <type>=2 bimolekulare nukleophile Substitution \mech[2] S<sub>N</sub>2
- <type>=se elektrophile Substitution \mech[se] S<sub>E</sub>
- <type>=1e unimolekulare elektrophile Substitution \mech[1e] S<sub>E</sub>1
- <type>=2e bimolekulare elektrophile Substitution \mech[2e] S<sub>E</sub>2
- <type>=ar elektrophile Substitution am Aromaten \mech[ar] Ar-S<sub>E</sub>
- <type>=e Eliminierung \mech[e] E
- <type>=1 unimolekulare Eliminierung \mech[e1] E1
- <type>=2 bimolekulare Eliminierung \mech[e2] E2
- <type>=cb unimolekulare Eliminierung „conjugated base“, also über Carbanion als Zwischenstufe \mech[cb] E1<sub>cb</sub>

Dieser Befehl ist auch in Mathematikumgebungen und im \ce-Befehl von ‘mhchem’ einsetzbar.

```
1 $\mech[cb]$\ce{\mech[2]}\ce{\mech[ar]}
```

E1<sub>cb</sub> S<sub>N</sub>2 Ar-S<sub>E</sub>

---

<sup>1)</sup> <http://www.ctan.org/pkg/chemfig>

## 9 Redoxreaktionen

‘chemmacros’ stellt zwei Befehle zur Verfügung<sup>1)</sup>, mit denen die Elektronenübertragungen bei Redoxreaktionen angezeigt werden können. Die Befehle verwenden ‘TikZ’.

```
1 \OX{<name>,<atom>}
2 \redox(<name1>,<name2>)[<tikz>][<dim>]{<text>}
```

Mit `\OX` wird ein `<atom>` in eine Node geschrieben, die den Namen `<name>` bekommt. Hat man zwei `\OX` gesetzt, können Sie durch `\redox` mit einer Linie verbunden werden. Dazu werden die Namen der zu verbindenden Nodes in die runden Klammern geschrieben. Da `\redox` ein Tikzpicture mit den Optionen `remember picture,overlay` zeichnet, muss das Dokument *mindestens zweimal kompiliert* werden.

```
1 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b){
    oxidation}
oxidation

$$\overbrace{\text{Na} \rightarrow \text{Na}^{\oplus}}$$

```

Die Linie kann via `<tikz>` mit ‘TikZ’-Keys angepasst werden:

```
1 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{
    ox}

$$\overbrace{\text{Na} \rightarrow \text{Na}^{\oplus}}^{\text{ox}}$$

```

NEU v.1.1

Mit `<dim>` kann die Länge der vertikalen Linie festgelegt werden. Diese ist per Default `.6 em` lang. Diese Default-Länge wird mit `<dim>` multipliziert. Gibt man hier einen negativen Wert an, wird die Linie *unterhalb* des Textes geschrieben.

```
1 \OX{a,Na} $\rightarrow$ \OX{b,Na}\pch
2 \redox(a,b)[->,red]{ox}
3 \redox(a,b)[<-,blue][-1]{red}

$$\overbrace{\text{Na} \rightarrow \text{Na}^{\oplus}}^{\text{ox}}$$


$$\underbrace{\phantom{\text{Na} \rightarrow \text{Na}^{\oplus}}}_{\text{red}}$$

```

Die Default-Länge der vertikalen Linien kann mit `\setredoxdist{<length>}` angepasst werden:

```
1 \OX{a,A} $\rightarrow$ \OX{b,B}
2 \redox(a,b){}
3 \bigskip
```

<sup>1)</sup> Dank an [Peter Cao](#), der das Feature vorschlug.

```

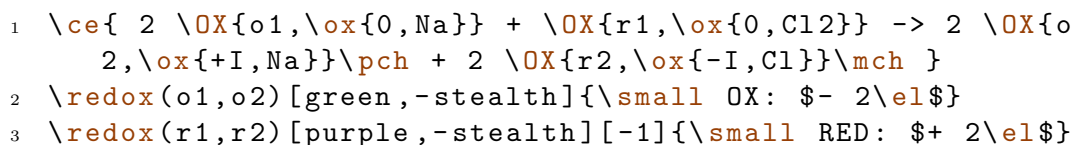
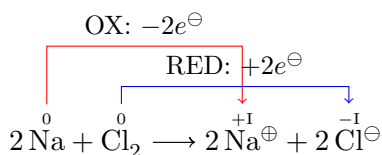
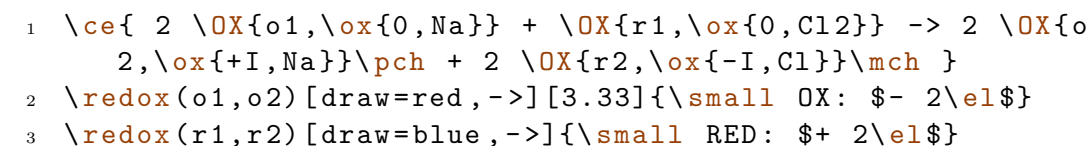
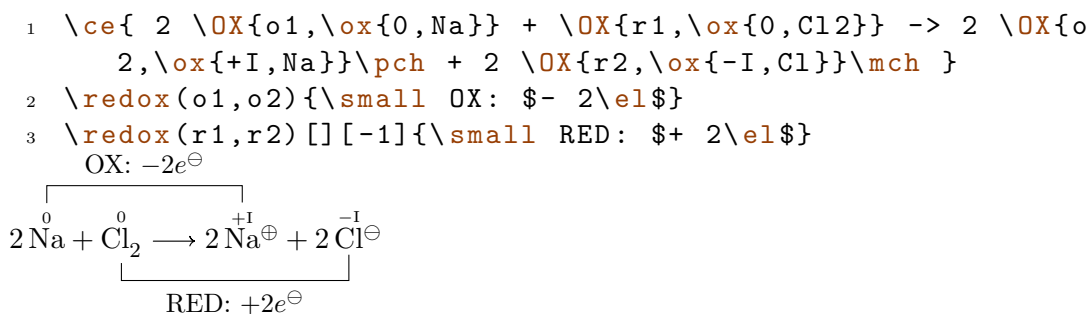
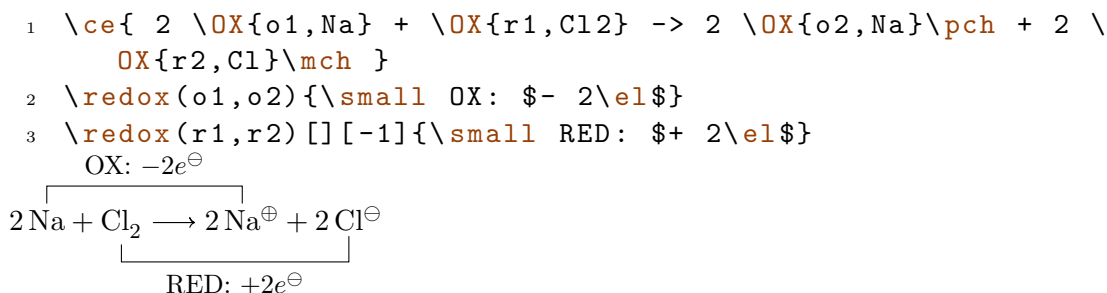
4 \setredoxdist{1em}
5 \OX{a,A} $\rightarrow$ \OX{b,B}
6 \redox(a,b){}

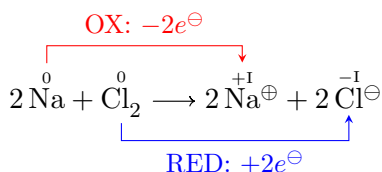
```



Mit einem leeren Argument wird die Voreinstellung wieder hergestellt.

Die Befehle funktionieren auch mit dem ‘mhchem’-Befehl `\ce` und in Verbindung mit dem `\ox`-Befehl (Abschnitt 7.2).





## 10 (Standard) Zustand, Thermodynamik

### 10.1 Thermodynamische Größen

Die folgenden Befehle verwenden ‘siunitx’:

- `\Enthalpy[<sub>,<sup>,<unit>,<subscript pos>]{<value>}`
- `\Entropy[<sub>,<sup>,<unit>,<subscript pos>]{<value>}`
- `\Gibbs[<sub>,<sup>,<unit>,<subscript pos>]{<value>}`

Ihr Einsatz ist eigentlich selbsterklärend:

mit `\Enthalpy{123}` wird  $\Delta H^{\ominus} = 123 \text{ kJ mol}^{-1}$  gesetzt.

Möchte man genauer spezifizieren, wofür die Enthalpie gelten soll (Reaktion, Bildung, ...), kann man das über das erste optionale Argument (<sub>) machen:

`\Enthalpy[r]{123}`  $\Delta_r H^{\ominus} = 123 \text{ kJ mol}^{-1}$

`\Enthalpy[Subl.,,right]{123}`  $\Delta H_{\text{Subl.}}^{\ominus} = 123 \text{ kJ mol}^{-1}$

NEU v.1.1

Im letzten Beispiel haben Sie auch gleich den Einsatz des vierten optionalen Arguments (<subscript pos>) gesehen: es bestimmt, ob das Subskript links oder rechts von Symbol eingesetzt wird und kann die Werte `left` (oder leer) oder `right` annehmen.

Das Standardzustandssymbol kann durch das zweite optionale Argument (<sup>) ersetzt werden...

`\Enthalpy[, \transitionstatesymbol]{123}`  $\Delta H^{\ddagger} = 123 \text{ kJ mol}^{-1}$

... und ist sonst abhängig davon, ob das Paket ‘chemstyle’ geladen wurde, siehe Abschnitt 10.2.

Mit dem dritten optionalen Argument (<unit>) schließlich kann die Einheit geändert werden:

`\Enthalpy[, , \kilo\calory\per\mole]{123}`  $\Delta H^{\ominus} = 123 \text{ kcal mol}^{-1}$

Die Einheit wird jeweils entsprechend der Regeln von ‘siunitx’ gesetzt und hängt von den dort gewählten Optionen ab.

- 1 `\Enthalpy{-1234.56e3}\par`
- 2 `\sisetup{per-mode=symbol,exponent-product=\cdot,output-decimal-marker={,},group-four-digits=true}`

```
3 \Enthalpy{-1234.56e3}
```

$$\Delta H^\ominus = -1234.56 \times 10^3 \text{ kJ mol}^{-1}$$

$$\Delta H^\ominus = -1\,234,56 \cdot 10^3 \text{ kJ/mol}$$

Ganz entsprechendes gilt für

```
1 \Entropy{12.3}, \Gibbs{-12.3}.
```

$$S^\ominus = 12.3 \text{ J K}^{-1} \text{ mol}^{-1}, \Delta G^\ominus = -12.3 \text{ kJ mol}^{-1}.$$

### 10.1.1 Neue Größen definieren

Mit dem Befehl

```
1 \setnewstate[<standard sup>,<Delta symbol>,<subscript pos>]
  <name>{<symbol>}{<unit>}
```

können Sie auch neue entsprechende Befehle festlegen:

```
1 \setnewstate{Helmholtz}{A}{\kilo\joule\per\mole}
2 \setnewstate[, ,right]{ElPot}{E}{\volt}
3 \Helmholtz{123.4} \ElPot{-1.1} \ElPot[\ce{Sn}|\ce{Sn \pch
  [2]}||\ce{Pb \pch[2]}|\ce{Pb},0]{0.01}
```

$$\Delta A^\ominus = 123.4 \text{ kJ mol}^{-1} \quad \Delta E = -1.1 \text{ V} \quad \Delta E_{\text{Sn}|\text{Sn}^{2\oplus}||\text{Pb}^{2\oplus}|\text{Pb}}^0 = 0.01 \text{ V}$$

Wie Sie sehen, hat `\ElPot` durch die Definition sein Subskript nun per Default rechts. Natürlich lässt es sich durch Angabe der Option `\ElPot[r,,,left]{0.12}` immer noch nach links setzen:  $\Delta_r E = 0.12 \text{ V}$  (auch wenn das in diesem Fall vielleicht wenig sinnvoll ist).

Tatsächlich wurden

```
1 \Enthalpy, \Entropy, \Gibbs
```

über

```
1 \setnewstate{Enthalpy}{H}{\kilo\joule\per\mole}
2 \setnewstate[, ]{Entropy}{S}{\joule\per\kelvin\per\mole}
3 \setnewstate{Gibbs}{G}{\kilo\joule\per\mole}
```

definiert.

### 10.1.2 Größen umdefinieren

NEU v.1.1

Mit

```
1 \renewstate[<standard sup>,<Delta symbol>,<subscript pos>]{<name>}{<symbol>}{<unit>}
```

können Sie die vordefinierten Befehle überschreiben:

```
1 \renewstate{Enthalpy}{h}{\joule}
2 \Enthalpy[f]{12.5}
```

$$\Delta_f h^\ominus = 12.5 \text{ J}$$

Um thermodynamischen Konventionen zu folgen, könnte man also eine molare und eine absolute Größe definieren:

```
1 \setnewstate[ ]{enthalpy}{h}{\kilojoule\per\mole}% molar
2 \renewstate[ ]{Enthalpy}{H}{\kilojoule}% absolut
3 \enthalpy{-12.3} \Enthalpy{-12.3}
```

$$\Delta h = -12.3 \text{ kJ mol}^{-1} \quad \Delta H = -12.3 \text{ kJ}$$

## 10.2 State

Die in Abschnitt 10.1 vorgestellten Befehle verwenden intern alle den Befehl

```
1 \State[<sup>,<Delta symbol>,<subscript pos>]{<symbol>}{<subscript>}
```

Mit diesem Befehl können die thermodynamischen Größen auch ohne Zahl und Einheit angegeben werden. Beachten Sie, dass `{<subscript>}` ein *optionales* Argument ist.

Beispiele:

```
1 \State{A}, \State{G}{f}, \State[ , ,right]{E}{\ce{Na}}, \
  State[\SI{1000}{\celsius}]{H}
```

$$\Delta A^\ominus, \Delta_f G^\ominus, \Delta E_{\text{Na}}, \Delta H^{1000^\circ\text{C}}$$

Zugegeben: nicht in jedem Fall ist das einfacher oder bequemer als die direkte Eingabe wie zum Beispiel `\Delta E_{\ce{Na}}`. Die Beispiele sollen lediglich verdeutli-



chen, wie der Befehl funktioniert und eingesetzt werden *kann*. Im ersten Fall spart man sicher Zeit gegenüber  $\Delta_{\mathrm{f}}G^{\circ}$ .

Das Standardzustandssymbol  $\ominus$  wird nur angezeigt, wenn das Paket ‘chemstyle’ geladen wurde und mit dem Befehl `\standardstate` zur Verfügung steht. Sonst wird das Symbol `\circ` verwendet. `\State{A}{b}`: Wenn ‘chemstyle’ geladen wurde  $\Delta_{\mathrm{b}}A^{\ominus}$ , sonst  $\Delta_{\mathrm{b}}A^{\circ}$ .

NEU v.1.1

Mit dem Befehl

```
1 \setstatesubscript{<subscript pos>}
```

kann man die Voreinstellung der Subskript-Position festlegen. Wie Sie sicher schon festgestellt haben, ist sie per Default `left`.

```
1 \State{A}{b} \setstatesubscript{right}
2 \setstatesubscript{right}
3 \State{A}{b}
```



Diese Einstellung hat *keinen Einfluss* auf die Funktionsweise von `\setnewstate` und `\renewstate`.

## 11 Spektroskopie

NEU v.1.1

Wenn Substanzen darauf untersucht werden, ob sie sind, was sie sein wollen, kommt man in der Regel um die NMR-Spektroskopie nicht herum. Dabei werden die Messergebnisse oft in der Form  $^1\text{H-NMR}$  (400 MHz,  $\text{CDCl}_3$ ):  $\delta = 1.59$  (q, 1H,  $J = 11.6$  Hz, H-4)... angegeben. ‘chemmacros’ stellt einen Befehl zur Verfügung, der diese Angaben vereinfachen soll (verwendet ‘siunitx’).

```
1 \NMR{<num>,<elem>}<num>,<unit> [<solvent>]
2 \NMR*{<num>,<elem>}<num>,<unit> [<solvent>]
```

Alle Argumente dieses Befehls sind optional. In der einfachsten Eingabe liefert der Befehl:

- `\NMR`  $^1\text{H-NMR}$ :  $\delta$  (entspricht weitestgehend dem ‘bpchem’-Befehl `\HNMR`)
- `\NMR*`  $^1\text{H-NMR}$  (Ausgabe ohne :  $\Delta$ )

Mit dem ersten Argument lässt die die Art der NMR ändern:

- `\NMR{13,C}`  $^{13}\text{C}$ -NMR:  $\delta$
- `\NMR*{13,C}`  $^{13}\text{C}$ -NMR

Mit dem zweiten Argument kann die bei der Messung verwendete Frequenz (in MHz) angegeben werden:

- `\NMR(400)`  $^1\text{H}$ -NMR (400 MHz):  $\delta$
- `\NMR*(400)`  $^1\text{H}$ -NMR (400 MHz)

Die verwendete Einheit kann durch explizite Angabe geändert werden:

- `\NMR(4e8,\hertz)`  $^1\text{H}$ -NMR ( $4 \times 10^8$  Hz):  $\delta$
- `\NMR*(4e8,\hertz)`  $^1\text{H}$ -NMR ( $4 \times 10^8$  Hz)

Das Setup von ‘siunitx’ wirkt sich ebenfalls auf diesen Befehl aus:

- `\sisetup{exponent-product=\cdot}\NMR(4e8,\hertz)`  $^1\text{H}$ -NMR ( $4 \cdot 10^8$  Hz):  $\delta$
- `\sisetup{exponent-product=\cdot}\NMR*(4e8,\hertz)`  $^1\text{H}$ -NMR ( $4 \cdot 10^8$  Hz)

Mit dem dritten Argument schließlich kann der verwendete Standard spezifiziert werden:

- `\NMR[CDC13]`  $^1\text{H}$ -NMR ( $\text{CDCl}_3$ ):  $\delta$
- `\NMR*[CDC13]`  $^1\text{H}$ -NMR ( $\text{CDCl}_3$ )

Das Lösungsmittel wird abhängig davon, ob Sie ‘mhchem’ verwenden, in den `\ce`-Befehl gesetzt. Wenn Sie ‘mhchem’ nicht verwenden, wird der Index nicht automatisch erkannt und Sie müssen die Mathematik-Variante verwenden:

- `\NMR[CDC1$_3$]`  $^1\text{H}$ -NMR ( $\text{CDCl}_3$ ):  $\delta$
- `\NMR*[CDC1$_3$]`  $^1\text{H}$ -NMR ( $\text{CDCl}_3$ )

Alle Argumente können beliebig kombiniert werden, der Befehl kann außerdem auch im Mathematikmodus verwendet werden.

Wenn Sie wollen, dass das Erscheinungsbild mit dem von ‘bpchem’ übereinstimmt (vgl. ‘bpchem’-Befehl `\HNMR` <sup>1</sup>H-NMR:  $\delta$  und ‘chemmacros’-Befehl `\NMR` <sup>1</sup>H-NMR:  $\delta$ ), dann verwenden Sie die Paketooption `bpchem` (siehe Abschnitt 2).

Beispiele:

```
1 \NMR{13,C}(100) \\
2 \NMR*{13,C}(100) \\
3 \NMR*{19,F}[CFC13] \\
4 \NMR*{19,F}(285)[CFC13] \\
5 \NMR(400)[CDCl3] = \num{1.59} (q, 1H, \textit{J} = \SI
  {11.6}{\hertz}, H-4)
```

<sup>13</sup>C-NMR (100 MHz):  $\delta$

<sup>13</sup>C-NMR (100 MHz)

<sup>19</sup>F-NMR (CFC1<sub>3</sub>)

<sup>19</sup>F-NMR (285 MHz, CFC1<sub>3</sub>)

<sup>1</sup>H-NMR (400 MHz, CDCl<sub>3</sub>):  $\delta = 1.59$  (q, 1H,  $J = 11.6$  Hz, H-4)

## 12 Befehle für ‘mhchem’

Einige Makros sind für den Gebrauch mit ‘mhchem’ gedacht und stehen zur Verfügung, wenn das Paket geladen wird. Bevor sie beschrieben werden einige Worte zur Verwendung von Befehlen innerhalb der `\ce-` und `\cee-`-Befehle. Wohl aufgrund der Art und Weise, wie die Befehle ausgelesen werden, kann es immer wieder zu Schwierigkeiten kommen, vor allem, wenn Befehle mit Argumenten dort eingesetzt werden.

Häufig müssen zum Beispiel Leerstellen gelassen werden:

```
1 \ce{Na\pch} \\ % kein Problem
2 \ce{Ca\pch[2]} \\ % falsche Darstellung
3 \ce{Ca \pch[2]} \\ % richtige Darstellung
4 \ce{Ca$\pch[2]$} % richtige Darstellung
```

Na<sup>⊕</sup>

Ca<sup>⊕</sup>[<sub>2</sub>]

Ca<sup>2⊕</sup>

Ca<sup>2⊕</sup>

In der Regel ist es auch nötig, Argumente mit geschweiften Klammern enden zu lassen:

```
1 \ce{\mch OMe} \\ % falsche Darstellung
```

```

2 \ce{\mch{} OMe}\\ % richtige Darstellung
3 \ce{\$ \mch$ OMe} % richtige Darstellung

```

$\ominus OMe$

$\ominus OMe$

$\ominus OMe$

Das betrifft *nicht* nur ‘chemmacros’-Befehle!

```

1 \ce{A \quad B} \ce{Na2\textbf{O}}\\ % falsche
   Darstellung
2 \ce{A \quad{ } B} \ce{Na2 \textbf{O}}\\ % richtige
   Darstellung
3 \ce{A \$\quad$ B} \ce{Na2 \textbf{O}} % richtige
   Darstellung

```

A B Na<sub>2</sub>O

A B Na<sub>2</sub>**O**

A B Na<sub>2</sub>**O**

Wie Sie sehen können, können Sie in den meisten Fällen anstelle der Leerstellen oder geschweiften Klammern die entsprechenden Befehle zwischen \$ \$ schreiben.

## 12.1 Reaktionsumgebungen

### 12.1.1 Durch ‘chemmacros’ definiert

Es stehen die Umgebungen...

```

1 \begin{reaction}
2   <mhchem code>
3 \end{reaction}
4 \begin{reactions}
5   <mhchem code>
6 \end{reactions}

```

...sowie die Varianten...

```

1 \begin{reaction*}
2   <mhchem code>
3 \end{reaction*}
4 \begin{reactions*}

```

```

5     <mhchem code>
6 \end{reactions*}

```

...zur Verfügung. Mit ihnen lassen sich (un)nummerierte Reaktionsgleichungen ähnlich mathematischer Formeln erstellen.

Die Umgebung `reaction/reaction*` verwendet die Mathematik-Umgebung `equation/equation*` und die Umgebung `reactions/reactions*` die Mathematik-Umgebung `align/align*`, um die Reaktionen zu setzen.

Reaktion mit Zähler:

```

1 \begin{reaction}
2   A -> B
3 \end{reaction}

```



Reaktion ohne Zähler:

```

1 \begin{reaction*}
2   C -> D
3 \end{reaction*}

```



mehrere ausgerichtete Reaktionen mit Zähler:

```

1 \begin{reactions}
2   A      &-> B + C \\
3   D + E &-> F
4 \end{reactions}

```

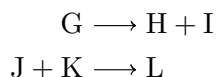


mehrere ausgerichtete Reaktionen ohne Zähler:

```

1 \begin{reactions*}
2   G      &-> H + I \\
3   J + K &-> L
4 \end{reactions*}

```



Wenn Sie das Aussehen des Zählers ändern wollen, so können Sie `\renewtagform{<tagname>}[<format>]{<right delim>}{<left delim>}`<sup>1)</sup> verwenden.

```
1 \renewtagform{CMreaction}[R \textbf]{[]{} }
2 \begin{reaction}
3   H2O + CO2 <=> H2CO3
4 \end{reaction}
```



### 12.1.2 Eigene Reaktionen

NEU v.1.1

Sie können auch neue Umgebungen selbst festlegen:

```
1 \newreaction{<name>}{<math name>}
```

Mit `<name>` legen Sie den Namen der neuen Umgebung fest. Bei `<math name>` wählen Sie die zu verwendende Mathematik-Umgebung aus.

Der Befehl hat zwei Varianten. Zum einen `\newreaction*`. Damit wird auch die entsprechende Sternvariante definiert, vorausgesetzt, die gesternte Mathematik-Umgebung existiert. Wenn nicht, wird es einen Fehler geben.

Zum anderen `\newreaction+`. Damit wird eine Umgebung mit Argument zu erstellt. Auch hier gilt: die entsprechende Mathematikumgebung muss bereits ein Argument haben.

Auch die Kombination `\newreaction*+` kann verwendet werden.

Die beiden bestehenden Umgebungen sind über...

```
1 \newreaction*{reaction}{equation}
2 \newreaction*{reactions}{align}
```

...definiert.

---

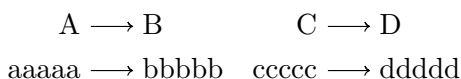
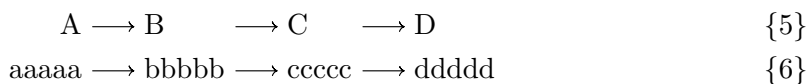
<sup>1)</sup> Durch das Paket 'mathtools' zur Verfügung gestellt

Nehmen wir einmal an, Sie möchten die Ausrichtungsmöglichkeiten der `alignat`-Umgebung auch für ‘mhchem’-Reaktionen nutzen. Sie könnten folgendes tun:

```
1 \newreaction*+{reactionsat}{alignat}
```

Damit steht nun die Umgebung `reactionsat` zur Verfügung.

```
1 \begin{reactionsat}{3}
2   A      &-> B      &&-> C      &&-> D  \\\
3   aaaaa &-> bbbbb &&-> ccccc &&-> ddddd
4 \end{reactionsat}
5 \begin{reactionsat*}{2}
6   A      &-> B      & C      &-> D  \\\
7   aaaaa &-> bbbbb &\quad{} &-> ccccc &-> ddddd
8 \end{reactionsat*}
```



## 12.2 Phasen

NEU v.1.1

Folgende Befehle sind für Phasenangaben gedacht. Auch wenn diese Befehle für den Einsatz mit ‘mhchem’ gedacht waren, können sie genauso gut ohne verwendet werden und stehen auch zur Verfügung, wenn ‘mhchem’ nicht geladen wurde.

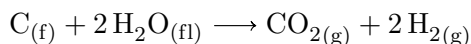
```
1 \solid[<anything>] % fest
2 \liquid[<anything>] % flüssig
3 \gas % gasförmig
4 \aq % gelöst in Wasser
```

Mir war es immer zu umständlich, die Phasenangaben ausschreiben zu müssen:

```
1 \ce{C_{(f)} + 2 H2O_{(f1)} -> CO2_{(g)} + 2 H2_{(g)}}
```

Dasselbe Ergebnis erhalten Sie daher nun mit

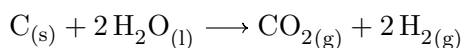
```
1 \ce{C\solid{} + 2 H2O\liquid{} -> CO2\gas{} + 2 H2\gas{}}
```



Der Vollständigkeit halber: `\NaCl\aq` gibt  $\text{NaCl}_{(\text{aq})}$ .

Wenn Sie lieber die englischen Bezeichnungen wollen, müssen Sie entweder auf die Paketoption `german` (siehe Abschnitt 2) verzichten, oder die Buchstaben als Argumente angeben:

```
1 \ce{C \solid[s] + 2 H2O \liquid[l] -> CO2\gas{ } + 2 H2\gas{ }}
```



Natürlich ist es egal, welchen Befehl mit Argument Sie verwenden, da beide lediglich ein Subskript mit Klammern erzeugen. `\solid[s]` ist identisch mit `\liquid[s]`.

Auch andere Einsätze sind denkbar:

```
1 C\solid[Graphit]
```



## 12.3 Beschriftungen

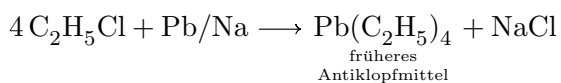
NEU v.1.1

Ein Punkt, der bei ‘mhchem’ bislang ein wenig umständlich war: Beschriften von Molekülen. Dafür stellt ‘chemmacros’ nun einen Befehl zur Verfügung:

```
1 \mhName[<width>][<textattr>]{<mhchem code>}{<name>}
```

Zum Beispiel:

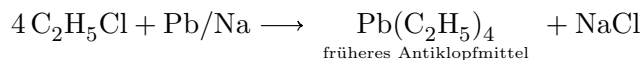
```
1 \ce{4 C2H5Cl + Pb / Na -> \mhName{Pb(C2H5)4}{früheres
Antiklopfmittel} + NaCl}
```



Wie Sie sehen, wird der Text per Default zentriert und `\tiny` unter das Molekül gesetzt, das Molekül nimmt dabei seinen normalen Raum ein. Sie können mit dem ersten optionalen Argument die Breite aber beliebig wählen.

```
1 \ce{4 C2H5Cl + Pb / Na -> \mhName[3cm]{Pb(C2H5)4}{früheres
Antiklopfmittel} + NaCl}
```

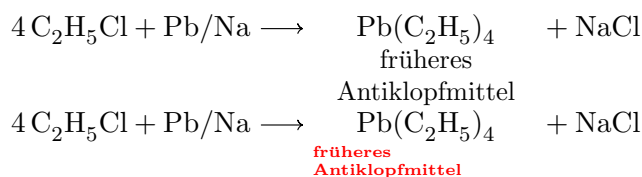




Die Textattribute sind in der Voreinstellung mit `\centering\tiny` festgelegt. Sie werden durch Eingabe anderer Attribute direkt zum Text nicht oder nur bedingt überschrieben:

```

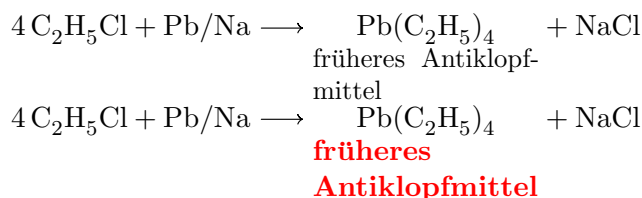
1 \ce{4 C2H5Cl + Pb / Na -> \mhName[3cm]{Pb(C2H5)4}{\small
    früheres Antiklopfmittel} + NaCl}\
2 \ce{4 C2H5Cl + Pb / Na -> \mhName[3cm]{Pb(C2H5)4}{\
    raggedright\color{red}\bfseries früheres
    Antiklopfmittel} + NaCl}
```



Durch das zweite optionale Argument lassen sie sich komplett überschreiben:

```

1 \ce{4 C2H5Cl + Pb / Na -> \mhName[3cm][\small]{Pb(C2H5)
    4}{früheres Antiklopfmittel} + NaCl}\
2 \ce{4 C2H5Cl + Pb / Na -> \mhName[3cm][\raggedright\color
    {red}\bfseries]{Pb(C2H5)4}{früheres Antiklopfmittel} +
    NaCl}
```



Um die Voreinstellung global zu ändern, gibt es den Befehl

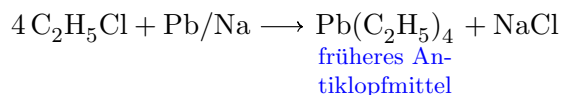
```

1 \setmhName{<textattr>}
```

Damit können Sie die Voreinstellung nach Ihren Vorlieben festlegen:

```

1 \setmhName{\centering\footnotesize\color{blue}}
2 \ce{4 C2H5Cl + Pb / Na -> \mhName{Pb(C2H5)4}{früheres
    Antiklopfmittel} + NaCl}
```



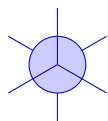
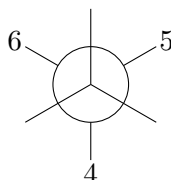
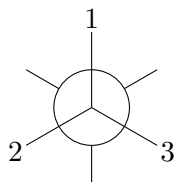
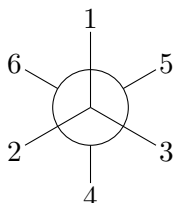
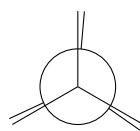
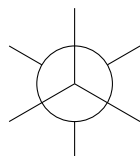
## 13 Newman-Projektionen

Mit dem Befehl

```
1 \newman[<angle>,<scale>,<tikz>]{<1>,<2>,<3>,<4>,<5>,<6>}
```

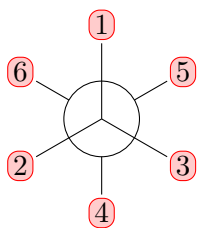
lassen sich Newman-Projektionen erstellen (verwendet 'TikZ'). Beispiele:

```
1 \newman{}\par% Standard: gestaffelt
2 \newman[175]{}\par% gedreht um 175 Grad => ekliptisch
3 \newman{1,2,3,4,5,6} \newman{1,2,3} \newman{,,4,5,6}\par
   % mit Atomen
4 \newman[,.75,draw=blue,fill=blue!20]{}% verkleinert und
   mit TikZ angepasst
```



Mit einer weiteren Option können die Nodes der Atome gestaltet werden:

```
1 \newman[][<tikz nodes>]{<1>,<2>,<3>,<4>,<5>,<6>}
2 % Beispiel:
3 \newman[][draw=red,fill=red!20,inner sep=2pt,rounded
   corners]{1,2,3,4,5,6}
```

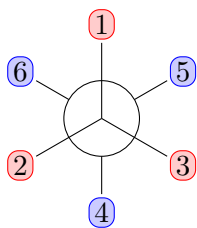


Wenn Sie die „vorderen“ und „hinteren“ Nodes verschieden gestalten wollen, können Sie noch eine dritte Option verwenden:

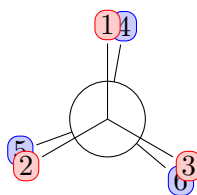
```
1 \newman[<tikz front nodes>][<tikz back nodes>]{<1>,<2>,<3>,<4>,<5>,<6>}
```

Beispiele:

```
1 \newman[draw=red,fill=red!20,inner sep=2pt,rounded corners][draw=blue,fill=blue!20,inner sep=2pt,rounded corners]{1,2,3,4,5,6}
```



```
1 \newman[170][draw=red,fill=red!20,inner sep=2pt,rounded corners][draw=blue,fill=blue!20,inner sep=2pt,rounded corners]{1,2,3,4,5,6}
```




## 14 p-Orbitale

‘chemmacros’ stellt auch Befehle zur Verfügung, mit denen p-Orbitale visualisiert werden können.

```

1 \porb[<size factor>,<color>,<angle>]
2 \phorb[<size factor>,<color>,<angle>]
3 \setorbheight{<length>}

```

Damit wird ein liegendes oder um <angle> gedrehtes p-Orbital dargestellt: `\porb \quad \porb[, ,30]` 

`\phorb` stellt nur ein halbes Orbital dar: `\phorb[,red,90]` 

Die Größe der dargestellten Orbitale hängt von einer internen Länge ab, die mit `\setorbheight{<length>}` eingestellt werden kann und die per Default den Wert `1em` hat.

```

1 \porb\par
2 \setorbheight{2em}\porb

```



Die Größe kann für ein einzelnes Orbital auch mit dem optionalen Argument <size factor> festgelegt werden.

```

1 \porb\par
2 \porb[2]\par
3 \porb[.5]

```

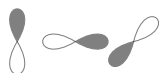


Mit den folgenden drei Befehlen stehen die in *x*-, *y*- und *z*-Richtung gedrehten Orbitale zusätzlich als Makro zur Verfügung:

```

1 \pzorb \quad \pyorb \quad \pxorb

```



Da die Orbitale als Tikzpicture mit Option `overlay` gezeichnet werden, werden sie alle an dieselbe Stelle gesetzt, wenn man sie nicht verschiebt:

```

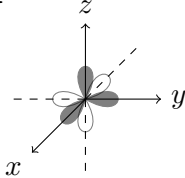
1 \hspace{2cm}\pxorb\pyorb\pzorb

```

```

2 \tikz[overlay]{
3   \draw[->](0,0)--(1,0)node[right]{$y$};
4   \draw[dashed](0,0)--(-1,0);
5   \draw[->](0,0)--(0,1)node[above]{$z$};
6   \draw[dashed](0,0)--(0,-1);
7   \draw[->](0,0)--(-.707,-.707)node[below left]{$x$};
8   \draw[dashed](0,0)--(.707,.707);
9 }

```

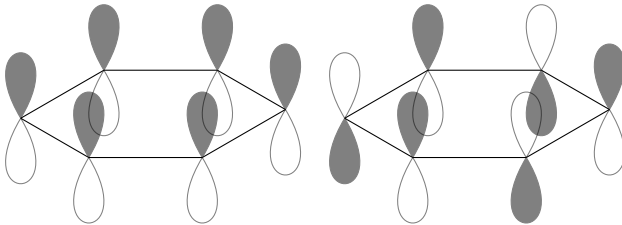


Die Orbitale lassen sich z. B. auch zusammen mit ‘chemfig’ einsetzen:

```

1 \setorbheight{2em}\setbondoffset{0pt}
2 \chemfig{?\pzorb-[ ,1.3]\pzorb-[:30,1.1]\pzorb-[:150,.9]\
   pzorb-[4,1.3]\pzorb-[: -150,1.1]\pzorb?}\quad
3 \chemfig{?\pzorb-[ ,1.3]{\porb[ ,,-90]}-[:30,1.1]\pzorb
   -[:150,.9]{\porb[ ,,-90]}-[4,1.3]\pzorb-[: -150,1.1]{\
   porb[ ,,-90]}?}

```



## 15 Liste der Befehle

<code>\el, \prt, \ntr, \HtO, \water, \Hpl, \Hyd, \Nu, \El, \transitionstatesymbol, \R</code>	Abschnitt 3: Teilchen, Ionen und ein Symbol
<code>\cip, \Rcip, \Scip, \Dfi, \Lfi, \E, \Z, \cis, \trans, \Rconf, \Sconf, \ortho, \meta, \para, \insitu, \abinitio</code>	Abschnitt 4: Stereo-Deskriptoren, Nomenklatur, lateinische Ausdrücke
<code>\pH, \pOH, \pKa, \pKb</code>	Abschnitt 6: Säure/Base
<code>\delm, \delp, \mch, \pch, \ox, \scrm, \scrp</code>	Abschnitt 7: Oxidationszahlen und (echte) Ladungen
<code>\mech</code>	Abschnitt 8: Reaktionsmechanismen
<code>\redox, \OX</code>	Abschnitt 9: Redoxreaktionen
<code>\Enthalpy, \Entropy, \Gibbs, \setnewstate, \renewstate, \State, \setstatesubscript</code>	Abschnitt 10: (Standard) Zustand, Thermodynamik
<code>\NMR</code>	Abschnitt 11: Spektroskopie
<code>\begin{reaction}, \begin{reaction*}, \begin{reactions}, \begin{reactions*}, \newreaction, \solid, \liquid, \gas, \mhName, \setmhName</code>	Abschnitt 12: Befehle für ‘mhchem’
<code>\newman</code>	Abschnitt 13: Newman-Projektionen
<code>\phorb, \porb, \pxorb, \pyorb, \pzorb, \setorbheight</code>	Abschnitt 14: p-Orbitale