

chemmacros v2.0a

2011/11/03

Clemens NIEDERBERGER

<http://www.mychemistry.eu/>
contact@mychemistry.eu

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

Inhaltsverzeichnis

1 Was ist neu?	3
2 Lizenz, Voraussetzungen	3
3 Paket-Optionen	3
4 Setup	4
5 Teilchen, Ionen und Symbole	5
6 Stereo-Deskriptoren, Nomenklatur, lateinische Ausdrücke	6
6.1 Stereo-Deskriptoren und Nomenklatur	6
6.2 Lateinische Ausdrücke	7
7 Einheiten für siunitx	8
8 Säure/Base	8
9 Oxidationszahlen, formale und echte Ladungen	9
9.1 Ladungen	9
9.2 Oxidationszahlen	10
9.3 Partialladungen und Ähnliches	11
10 Reaktionsmechanismen	12
11 Redoxreaktionen	12
12 (Standard) Zustand, Thermodynamik	15
12.1 Thermodynamische Größen	15
Neue Größen definieren	16
Größen undefinieren	16
12.2 State	17
13 Spektroskopie	18
14 Befehle für mhchem	19
14.1 Reaktionsumgebungen	20
Durch chemmacros definiert	20
Eigene Reaktionen	21
14.2 Phasen	22
14.3 Beschriftungen	23
15 Newman-Projektionen	24
16 s-, p- und Hybrid-Orbitale	26
17 Schlüssel-Übersicht	29
18 Vorschläge und Bugreports	30

1 Was ist neu?

Mit dem Update auf v2.0 hat sich im Hintergrund einiges geändert. Die Anpassung der Befehle wird nun komplett über ein Schlüssel/Wert-System vorgenommen. Dadurch gibt es jetzt deutlich mehr Möglichkeiten der Personalisierung von Befehlen. Das bedeutet auch, dass sich bei einigen Befehlen die Syntax verändert hat.

Um dennoch zu gewährleisten, dass Dokumente, die mit v1.* gesetzt wurden, weiterhin kompilieren, gibt es die Paket-Option `version=1`, die die alten Definitionen wiederherstellt. Bei Befehlen, die das betrifft wird wie rechts im Rand beispielhaft gezeigt auf die Änderung hingewiesen. Manche Befehle gibt es nun nicht mehr, doch auch sie werden durch die Paketoption `version=1` wiederhergestellt. Auf sie wird ebenfalls am Rand und im Text hingewiesen.

Der Befehl `\Befehl` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

`\Befehl` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

2 Lizenz, Voraussetzungen

chemmacros v2.0 steht unter der L^AT_EX Project Public License Version 1.3 oder später. (<http://www.latex-project.org/lppl.txt>)

chemmacros ruft intern die Pakete `expl3`, `xparse`¹, `l3keys2e`² und `xfrac`³ auf, die durch die Bundles `l3kernel`⁴ und `l3packages`⁵ bereitgestellt werden. `expl3` ist Teil des `l3kernel`-Bundles und `xparse`, `l3keys2e` und `xfrac` sind Teil des `l3packages`-Bundles.

Weiterhin lädt chemmacros die Pakete `siunitx`⁷, `mhchem`⁸, `mathtools`⁹ und `environ`¹⁰ sowie `TikZ` (`pgf`¹¹) und die `TikZ`libraries `calc` und `arrows`.

Für die Paketoption `bpchem` (Abschnitt 3) wird zusätzlich `bpchem`¹² benötigt und für die Option `xspace` das Paket `xspace`¹³.

¹ CTAN: `xparse`

² CTAN: `l3keys2e`

³ CTAN: `xfrac`

⁴ CTAN: `l3kernel`

⁵ CTAN: `l3packages`

⁶ CTAN: `expl3`

⁷ CTAN: `siunitx`

⁸ CTAN: `mhchem`

⁹ CTAN: `mathtools`

¹⁰ CTAN: `environ`

¹¹ CTAN: `pgf`

¹² CTAN: `bpchem`

¹³ CTAN: `xspace`

3 Paket-Optionen

chemmacros hat mehrere Paketoptionen. Sie alle können nach dem Schlüssel/Wert-Prinzip auf die Art `\usepackage[option1 = <value1>, option2 = <value2>]{chemmacros}` angegeben werden. Manche können auch ohne Wert aufgerufen werden (`\usepackage[option1, option2]{chemmacros}`). Sie entsprechen dann dem unterstrichenen Wert.

`bpchem = true/false` 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. (Default: `false`)

`circled = formal/all/none` chemmacros verwendet zwei verschiedene Formen von Ladungen¹⁴, die den Gebrauch von realen (+/−) und formalen (\oplus/\ominus) Ladungen widerspiegeln sollen. Mit der Option `formal` unterscheidet chemmacros zwischen ihnen, mit der Option `none` werden sie alle ohne Umkreisung dargestellt, mit `all`, werden alle umkreist. (Default: `formal`)

Dieses Dokument verwendet chemmacros mit folgenden Optionen:
`\usepackage[xspace,german]{chemmacros}`

`bpchem = false` ¹H-NMR: δ ;
`bpchem = true` ¹H-NMR: δ ;

`circled = none` − + − +
`circled = formal`, `circletype = chem` − + \ominus \oplus
`circled = all`, `circletype = chem` \ominus \oplus \ominus \oplus
`circled = formal`, `circletype = math` − + \ominus \oplus
`circled = all`, `circletype = math` \ominus \oplus \ominus \oplus

¹⁴ Dank an Christoph Schäfer, der mich auf die unpräzise Behandlung der Ladungen in v1.1 hinwies!

`circletype = chem/math` Diese Option schaltet zwischen zwei Darstellungsformen der umkreisten Ladungen hin und her: `\fplus` \oplus und `\oplus` \oplus . (Default: `chem`)

`EZ = chemmacros/cool` Der Befehl `\E` wird sowohl durch das Paket `cool`¹⁵ als auch `chemmacros` definiert. Mit der Option `EZ = cool` wird er durch `chemmacros` nicht definiert und stattdessen ein anderer Name verwendet, siehe Seite 7. (Default: `chemmacros`)

¹⁵ CTAN: `cool`

`german = true/false` Die Option ändert die Befehle `\pKa`, `\sld` und `\lqd` (Default: `false`)

`german = false` $pK_{A, (s), (l)}$
`german = true` $pK_{S, (f), (fl)}$

`version = 1/2` Die Option lädt die alte Definition einiger Befehle, damit Dokumente, die mit `v1.*` gesetzt wurden, nicht überarbeitet werden müssen. Im Rand wird bei den entsprechenden Befehlen mit neuer Syntax darauf hingewiesen. (Default: 2)

`xspace = true/false` Mit der Option bekommen die meisten Befehle ein `\xspace` angehängt. (Default: `true`)

4 Setup

Eine ganze Reihe von Befehlen haben Schlüssel/Wert-Paare erhalten, mit denen sie angepasst werden können. In der Regel können diese als Argument der Befehle direkt verwendet werden. Genauso können sie aber in der Regel durch den Befehl `\chemsetup` gesetzt werden.

`\chemsetup[<module>]{<key> = <value>}` oder
`\chemsetup{<module>/<key> = <value>}`

Die Schlüssel gehören jeweils einem Modul an, das bestimmt, für welche Befehle sie gelten. Wenn ein Schlüssel vorgestellt wird, können Sie im Rand die Eigenschaften sehen. Sie haben zwei Möglichkeiten, die Schlüssel mit dem `\chemsetup`-Befehl aufzurufen, wie im Kasten zu sehen.

Die Paketoptionen sind ebenso als Schlüssel mit `\chemsetup` setzbar. Sie gehören dem Modul `option` an.

`<key> = <value>`
 Default: `<default>`
 Modul: `<module>`

```

1 \chemsetup[option]{circled=none}\mch\ \pch\ \fmch\ \
   fpch\ \el\ \prt \
2 \chemsetup[option]{circled=formal}\mch\ \pch\ \fmch\ \
   fpch\ \el\ \prt \
3 \chemsetup[option]{circletype=math}\mch\ \pch\ \fmch\ \
   fpch\ \el\ \prt \
4 \chemsetup{option/circletype=chem,option/circled=all}\
   mch\ \pch\ \fmch\ \fpch\ \el\ \prt \
5 \chemsetup{option/circletype=math}\mch\ \pch\ \fmch\ \
   fpch\ \el\ \prt

```



Schlüssel, die *keinem* Modul angehören, können *nicht* mit `\chemsetup` gesetzt werden.

5 Teilchen, Ionen und Symbole

Eine Reihe einfacher Makros, um häufig gebrauchte Teilchen sowie ein Symbol darzustellen. Beachten Sie, dass die Darstellung von den verwendeten Paketoptionen abhängt, siehe Abschnitt 3.

- `\Hp1` H^+ (Proton)
- `\Hyd` OH^- (Hydroxid)
- `\Ht0` H_3O^+ (Oxonium) (**H** three **O**)
- `\water` H_2O
- `\el` e^- (Elektron)
- `\prt` p^+ (Proton)
- `\ntr` n^0 (Neutron)
- `\Nu` Nu^- (Nucleophil)
- `\El` E^+ (Elektrophil)
- `\ba` ba^- (Base)
- `\fplus` \oplus
- `\fminus` \ominus
- `\transitionstatesymbol` \neq Übergangszustandssymbol (verwendet `TikZ`)
- `\standardstate` \neq . Dieses Symbol wird durch `chemmacros` bereitgestellt, falls das Paket `chemstyle`¹⁶ nicht geladen wird, bei dem die Idee ausgeliehen wurde¹⁷.

Diese Befehle sind sowohl im Text- wie im Mathematikmodus einsetzbar.

¹⁶ CTAN: `chemstyle`

¹⁷ Dank an den Paketautoren **Joseph Wright**.

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

$$\text{\R[<sign>]{<subscript>}}$$

$$\text{R}_{\text{tert}}^+ \text{R}_{\text{sek}}^- \text{R}_{\text{prim}}$$

```
1 \R[+]{tert} \R[-]{sek} \R
   {prim}
```

Die beiden Teilchen `\Nu` und `\ba` können modifiziert werden. Dafür gibt es den Schlüssel `elpair`. Dieser Schlüssel hat nur Auswirkungen, wenn das Paket `chemfig`¹⁸ geladen wurde, da er dessen Befehl `\Lewis` verwendet.

```
elpair = false/dots/dash
Default: false
Modul: particle
```

$$\text{ba}^{\cdot-} \text{Nu}^{\cdot-}$$

```
1 \documentclass{article}
2 \usepackage{chemmacros,
   chemfig}
3 \begin{document}
4 \ba[elpair] \Nu[elpair=
   dash]
5 \end{document}
```

$$\text{ba}^{\cdot-} \text{Nu}^{\cdot-}$$

```
1 \documentclass{article}
2 \usepackage{chemmacros,
   chemfig}
3 \begin{document}
4 \chemsetup[particle]{
   elpair}
5 \ba \Nu
6 \end{document}
```

¹⁸ CTAN: [chemfig](#)

6 Stereo-Deskriptoren, Nomenklatur, lateinische Ausdrücke

6.1 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*) (`\E` wird z. B. auch durch das Paket `cool` definiert.
 Mit der Paketooption `EZ = cool` werden statt `\E` und `\Z` die
 Befehle `\Ent` und `\Zus` definiert.)
`\cis` *cis*
`\trans` *trans*

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.

- 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-Weins\ "aure = \cip{2S,3S}-Weins\ "aure \\
2 \Dfi-($-)-Threose = \cip{2S,3R}-($-)-2,3,4-
   Trihydroxybutanal \\
3 \cis-2-Buten = \Z-2-Butene, \cip{2E,4Z}-Hexadiene \\
4 \meta-Xylol = 1,3-Dimethylbenzene \\
5 % mit 'bpchem's Befehl \IUPAC:
6 \IUPAC{\Dfi\Wein\|s\ "aure} = \IUPAC{\cip{2S,3S}\Wein
   \|s\ "aure}, \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
cis-2-Buten = (*Z*)-2-Butene, (2*E*,4*Z*)-Hexadiene
m-Xylol = 1,3-Dimethylbenzene
 D-Weinsäure = (2*S*,3*S*)-Weinsäure, D-(−)-Threose = (2*S*,3*R*)-(−)-2,3,4-Trihydroxybutanal

Das Aussehen hängt natürlich von gewählten Font ab:

(2*S*,3*R*) (*E*)(*Z*)_{DL}
 (2*S*,3*R*) (*E*)(*Z*)_{DL}
 (2*S*,3*R*) (*E*)(*Z*)_{DL}

6.2 Lateinische Ausdrücke

Zuletzt gibt es noch zwei Befehle für gebräuchliche lateinische Ausdrücke.

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

Falls ebenfalls das Paket `chemstyle` geladen wurde¹⁹, wird für die Definition der `chemstyle`-Befehl `\latin` verwendet. Das bedeutet, dass das Schriftbild dann durch die `chemstyle`-Einstellung `abbremph` beeinflusst wird:

<i>in situ, ab initio</i>	<code>1 \insitu, \abinitio\\</code>
in situ, ab initio	<code>2 \cstsetup{abbremph=false}</code>
	<code>3 \insitu, \abinitio</code>

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

7 Einheiten für siunitx

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` cm³
- `\molar` mol dm⁻³
- `\moLar` mol L⁻¹
- `\Molar` M
- `\MolMass` g mol⁻¹
- `\normal` N
- `\torr` torr

Die Einheiten `\cmc`, `\molar` und `\Molar` werden auch durch `chemstyle` definiert. Sie werden durch `chemmacros` nur definiert, wenn `chemstyle` nicht geladen wird.

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

8 Säure/Base

Einfache Darstellung von pH , pK_S ...

```
\pH pH
\pOH pOH
\pKa[<num>] \pKa pK_S, \pKa[1] pK_{S1}
\pKb[<num>] \pKb pK_B, \pKb[1] pK_{B1}
\p{<anything>} z. B. \p{K_w} pK_w
```

```
pK_S pK_{S1} pK_B pK_{B1}
pK_A pK_{A1} pK_B pK_{B1}
```

```
1 \pKa \pKa[1] \pKb \pKb
  [1]\\
2 \chemsetup[option]{german
  =false}
3 \pKa \pKa[1] \pKb \pKb[1]
```

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 3.

9 Oxidationszahlen, formale und echte Ladungen

chemmacros unterscheidet zwischen realen (+/−) und formalen (\oplus/\ominus) Ladungssymbolen, siehe auch Abschnitt 3. Alle Befehle für formale Symbole beginnen mit einem `f`.

9.1 Ladungen

Einfaches Darstellen von (echten) Ladungen:

`\pch[<number>]` positive Ladung (**p**lus + **ch**arge)
`\mch[<number>]` negative Ladung (**m**inus + **ch**arge)

$+$, Na^+ , Ca^{2+}
 $-$, F^- , S^{2-}

```
1 \pch, Na\pch, Ca\pch[2]\
2 \mch, F\mch, S\mch[2]
```

Entsprechendes Darstellen von formalen Ladungen:

`\fpch[<number>]` positive Ladung
`\fmch[<number>]` negative Ladung

\oplus \ominus $3\oplus$ $3\ominus$

```
1 \fpch\ \fmch\ \fpch[3] \
   fmch[3]
```

Es gibt einen Schlüssel, der das Verhalten der Ladungen beeinflusst.

- `append = true/false` hängt bei `true` die Ladung mit einer leeren Gruppe an.

`append = true/false`
 Default: `false`
 Modul: `charges`

Der Schlüssel hat folgende Auswirkung:

$\text{H}_{(\text{aq})}^+$ $\text{H}_{(\text{aq})}^+$
 $\text{H}_{(\text{aq})}^+$ $\text{H}_{(\text{aq})}^+$

```
1 \chemsetup[charges]{
   append=false}
2 \ce{H\pch\aq} \ce{H\aq\
   pch}
3
4 \chemsetup[charges]{
   append=true}
5 \ce{H\pch\aq} \ce{H\aq\
   pch}
```

Auch wenn das Verhalten meistens unerwünscht sein mag, gibt es Fälle, in denen es sinnvoll sein kann, etwa in Kombination mit dem im nächsten Abschnitt besprochenen `\ox`-Befehl:

$\text{H}_{(\text{aq})}^+$
 $\text{H}_{(\text{aq})}^+$

```
1 \chemsetup[charges]{
   append=false}
2 \ce{\ox{1,H}\pch\aq}
3
4 \chemsetup[charges]{
   append=true}
5 \ce{\ox{1,H}\pch\aq}
```

9.2 Oxidationszahlen

Setzen von Oxidationszahlen:

`\ox[<keyval>]{<number>,<atom>}` setzt <number> über <atom>
 <number> muss eine (rationale) Zahl sein!

Der Befehl `\ox` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.



Es gibt eine Reihe von Schlüssel/Wert-Paaren, die eingesetzt werden können, um den `\ox`-Befehl zu modifizieren.

- `parse = true/false` ermöglicht bei `false` eine beliebige Eingabe für <number>.
- `roman = true/false` wechselt die Darstellung von römischen zu arabischen Ziffern.
- `pos = top/super/side` schreibt bei `top` <number> über <atom>, bei `super` rechts oben an <atom> und bei `side` rechts neben <atom> in Klammern.
- `explicit-sign = true/false` gibt auch bei Eingabe ohne Vorzeichen ein + aus.
- `decimal-marker = comma/point` Auswahl des Dezimalzeichens für formale Oxidationszahlen wie $\overset{1.2}{\text{X}}$.

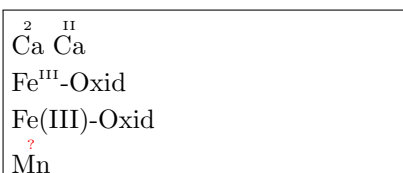
```
parse = true/false
Default: true
Modul: ox
```

```
roman = true/false
Default: true
Modul: ox
```

```
pos = top/super/side
Default: top
Modul: ox
```

```
explicit-sign = true/false
Default: false
Modul: ox
```

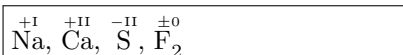
```
decimal-marker =
comma/point
Default: point
Modul: ox
```



Für die `pos=super`-Variante existiert noch der Shortcut `\ox*`:



Durch die Verwendung des `explicit-sign`-Schlüssels wird das Vorzeichen der Oxidationszahl immer angezeigt:



Im letzten Beispiel haben Sie gesehen, dass <atom> intern `mhchems \ce`-Befehl gesetzt wird. Allerdings liefert es nicht immer die besten Ergebnisse, diese Tatsache auszunutzen.

1 Vergleichen Sie O_2^{2-} mit O_2^{-1} oder O_2^{2-} .

Vergleichen Sie O_2^{2-} mit O_2^{-1} oder O_2^{2-} .

Manchmal benötigt man vielleicht formal gebrochene Oxidationszahlen wie 0.5 oder $\frac{1}{3}$. Das ist möglich:

1 $\text{Br}_2^{+0.5}$ $\text{I}_3^{+1/3}$ $\text{Br}_2^{+0.5}$ $\text{I}_3^{+1/3}$

$\text{Br}_2^{+0.5}$ $\text{I}_3^{+1/3}$ $\text{Br}_2^{+0.5}$ $\text{I}_3^{+1/3}$

9.3 Partialladungen und Ähnliches

Vielleicht selten gebraucht, nichtsdestoweniger nützlich:

δ^+ (delta + plus)
 δ^- (delta + minus)
 δ^\oplus
 δ^\ominus

Diese Makros können zum Beispiel mit dem `\ox`-Befehl eingesetzt werden oder zusammen mit dem Paket `chemfig`:

```
1 \chemsetup{
2   option/circled = all,
3   ox/parse      = false
4 }
5 \ox{\delp,H}\ox{\delm,Cl} \hspace*{1cm}
6 \chemfig{\chemabove[3pt]{\lewis{246,Br}}{\delm}-\chemabove[3pt]{H}{\delp}}
```

$\delta^\oplus \delta^\ominus$ δ^\ominus δ^\oplus
 HCl $\text{Br} - \text{H}$

Auch die folgenden Makros sind v. a. beim Einsatz mit `chemfig` nützlich.

+ (scriptstyle + plus)
 - (scriptstyle + minus)
 \oplus
 \ominus
 \oplus (im scriptscriptstyle)
 \ominus

```

1 \setatomsep{1.8em}\chemfig{CH_3-\chemabove{C}{\scrp
   }(-[6]C|H_3)-\vphantom{H_3}CH_3}
2
3 \chemfig{\fmch{}}|O-\chemabove{N}{\fscrp}{(-[1]O|\fmch)
   -[7]O|\fmch}

```

$$\begin{array}{c}
 \text{CH}_3 - \overset{+}{\text{C}} - \text{CH}_3 \\
 | \\
 \text{CH}_3 \\
 \diagup \quad \diagdown \\
 \ominus \text{O} - \text{N}^+ - \text{O}^\ominus
 \end{array}$$

10 Reaktionsmechanismen

Der Befehl

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

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

erlaubt die Angabe der verbreitetsten Reaktionsmechanismen. `<type>` kann verschiedene Werte annehmen:

- `\mech` (leer, kein opt. Argument) nukleophile Substitution S_N
- `\mech[1]` unimolekulare nukleophile Substitution S_N1
- `\mech[2]` bimolekulare nukleophile Substitution S_N2
- `\mech[se]` elektrophile Substitution S_E
- `\mech[1e]` unimolekulare elektrophile Substitution S_E1
- `\mech[2e]` bimolekulare elektrophile Substitution S_E2
- `\mech[ar]` elektrophile Substitution am Aromaten $Ar-S_E$
- `\mech[e]` Eliminierung E
- `\mech[e1]` unimolekulare Eliminierung $E1$
- `\mech[e2]` bimolekulare Eliminierung $E2$
- `\mech[cb]` unimolekulare Eliminierung „conjugated base“, also über Carbanion als Zwischenstufe $E1_{cb}$

11 Redoxreaktionen

`chemmacros` stellt zwei Befehle zur Verfügung²⁰, mit denen die Elektronenübertragungen bei Redoxreaktionen angezeigt werden können. Die Befehle verwenden `TikZ`.

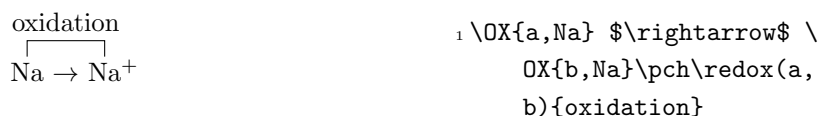
²⁰ Dank an [Peter Cao](#), der das Feature vorschlug.

```

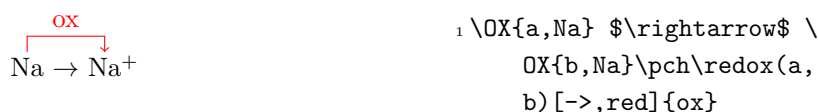
\OX{<name>,<atom>}
\redox(<name1>,<name2>)[<tikz>][<num>]{<text>}

```

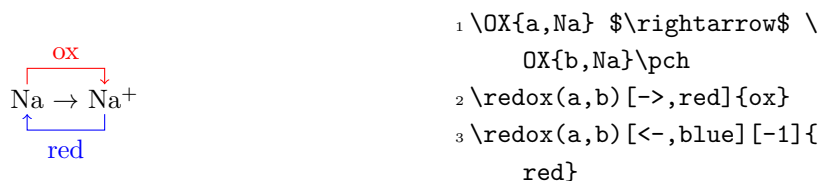
Mit `\OX` wird ein `<atom>` in eine Node geschrieben, die den Namen `<name>` bekommt. Hat man zwei `\OX` gesetzt, können Sie durch `\redox` mit einer Linie verbunden werden. Dazu werden die Namen der zu verbindenen Nodes in die Runden Klammern geschrieben. Da `\redox` ein `TikZpicture` mit den Optionen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



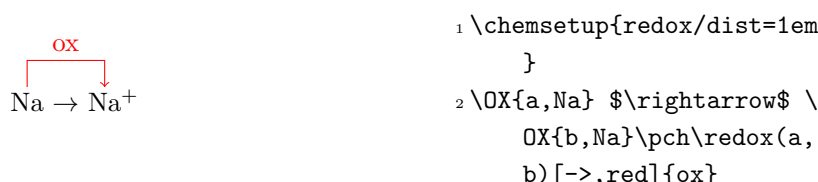
Die Linie kann mit dem Argument [`<tikz>`] mit **TikZ**-Keys angepasst werden:



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

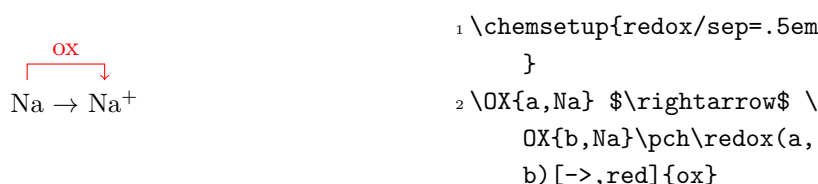


Die Default-Länge kann mit der Schlüssel `dist` geändert werden.



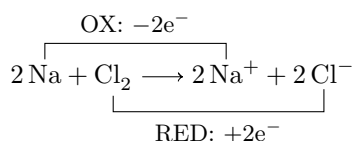
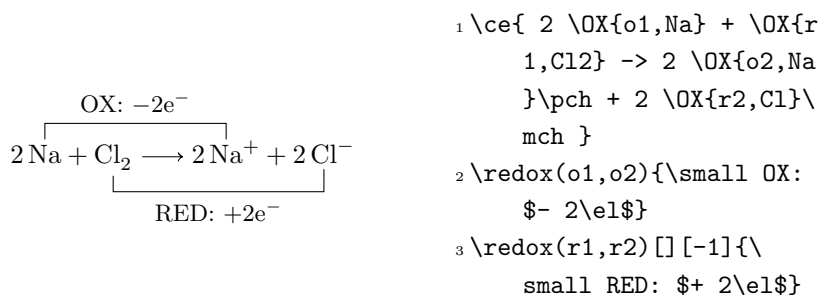
```
dist = <dim>  
Default: .6em  
Modul: redox
```

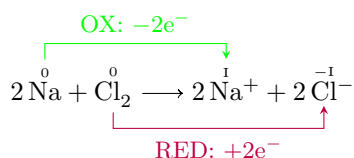
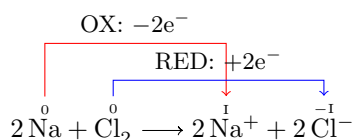
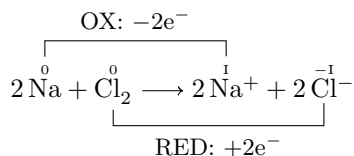
Zusätzlich kann mit dem Key `sep` der Abstand zwischen Linienbeginn und Atom geändert werden.



sep = <dim>
Default: .2em
Modul: redox

Beispiele:





```

1 \chemsetup[charges]{
  append}
2 \ce{ 2 \OX{o1,\ox{0,Na}}
  + \OX{r1,\ox{0,Cl2}}
  -> 2 \OX{o2,\ox{+1,Na}}
  }\pch + 2 \OX{r2,\ox
  {-1,Cl}}\mch }
3 \redox(o1,o2){\small OX:
  $- 2\el$}
4 \redox(r1,r2)[][-1]{\small
  RED: $+ 2\el$}

```

```

1 \chemsetup[charges]{
  append}
2 \ce{ 2 \OX{o1,\ox{0,Na}}
  + \OX{r1,\ox{0,Cl2}}
  -> 2 \OX{o2,\ox{+1,Na}}
  }\pch + 2 \OX{r2,\ox
  {-1,Cl}}\mch }
3 \redox(o1,o2)[draw=red
  ,->][3.33]{\small OX:
  $- 2\el$}
4 \redox(r1,r2)[draw=blue
  ,->]{\small RED: $+
  2\el$}

```

```

1 \chemsetup[charges]{
  append}
2 \ce{ 2 \OX{o1,\ox{0,Na}}
  + \OX{r1,\ox{0,Cl2}}
  -> 2 \OX{o2,\ox{+1,Na}}
  }\pch + 2 \OX{r2,\ox
  {-1,Cl}}\mch }
3 \redox(o1,o2)[green,-
  stealth]{\small OX:
  $- 2\el$}
4 \redox(r1,r2)[purple,-
  stealth][[-1]{\small
  RED: $+ 2\el$}

```

In v1.1 gab es noch den Befehl `\setredoxdist{<dim>}`. Mit Laden der Paketooption `version=1` wird er bereitgestellt.

`\setredoxdist{<dim>}` wird nicht mehr bereitgestellt. Verwenden Sie die Paketooption `version=1`, um ihn wieder zu reaktivieren.

12 (Standard) Zustand, Thermodynamik

12.1 Thermodynamische Größen

Die folgenden Befehle verwenden siunitx:

```
\Enthalpy[<keyval>](<subscript>){<value>}
\Entropy[<keyval>](<subscript>){<value>}
\Gibbs[<keyval>](<subscript>){<value>}
```

Ihr Einsatz ist eigentlich selbsterklärend:

$\Delta H^\ominus = 123 \text{ kJ mol}^{-1}$	1 \Enthalpy{123} \\\
$S^\ominus = 123 \text{ J K}^{-1} \text{ mol}^{-1}$	2 \Entropy{123} \\\
$\Delta G^\ominus = 123 \text{ kJ mol}^{-1}$	3 \Gibbs{123}

Das Argument (<subscript>) fügt ein Subskript zur Spezifikation hinzu: $\text{\Enthalpy(r)}\{123\} \Delta_r H^\ominus = 123 \text{ kJ mol}^{-1}$.

Mit mehreren Schlüsseln kann das Erscheinungsbild angepasst werden.

- exponent = <anything>
- delta = <anything>/false
- subscript = left/right
- unit = <unit>

Die Default-Einstellungen sind je nach Befehl unterschiedlich.

$\Delta H^\ominus = -285 \text{ kJ}$	1 \Enthalpy[unit=\kilo\joule]{-285} \\\
$G^\ominus = 0 \text{ kJ mol}^{-1}$	2 \Gibbs[delta=false]{0} \\\
$\Delta S = 56.7 \text{ J K}^{-1} \text{ mol}^{-1}$	3 \Entropy[delta=\Delta,exponent=]{56.7}

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

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

$\Delta H^\ominus = -1234.56 \times 10^3 \text{ kJ mol}^{-1}$
$\Delta H^\ominus = -1\,234,56 \cdot 10^3 \text{ kJ/mol}$

Der Befehl \Enthalpy hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

Der Befehl \Entropy hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

Der Befehl \Gibbs hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

exponent = <anything>
Default: \standardstate
kein Modul

delta = <anything>/false
kein Modul

subscript = left/right
kein Modul

unit = <unit>
kein Modul

Neue Größen definieren

Mit dem Befehl

```
\setnewstate[<keyval>]{<name>}{<symbol>}{<unit>}
```

können Sie auch neue entsprechende Befehle festlegen:

$$\Delta A^\ominus = 123.4 \text{ kJ mol}^{-1}$$

$$\Delta E = -1.1 \text{ V}$$

$$\Delta E_{\text{Sn}|\text{Sn}^{2+}||\text{Pb}^{2+}|\text{Pb}}^0 = 0.01 \text{ V}$$

```
1 \setnewstate{Helmholtz}{A
   }\kilo\joule\per\
   mole}
2 \setnewstate[subscript-
   left=false,exponent
   =]{ElPot}{E}{\volt}
3 \Helmholtz{123.4} \
4 \ElPot{-1.1} \
5 \ElPot[exponent=0] ($\ce{
   Sn}\ce{Sn \pch
   [2]}||\ce{Pb \pch
   [2]}|\ce{Pb}$){0.01}
```

Der Befehl `\setnewstate` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

`exponent = <anything>`
Default: `\standardstate`
kein Modul

`delta = <anything>/false`
Default: `\Delta`
kein Modul

`subscript-left = true/false`
Default: `true`
kein Modul

`subscript = <anything>`
kein Modul

Der Befehl hat einige Schlüssel, mit denen die Default-Einstellungen des neuen Befehls festgelegt werden können.

- `exponent = <anything>`
- `delta = <anything>/false`
- `subscript-left = true/false`
- `subscript = <anything>`

Größen umdefinieren

Mit

```
\renewstate[<keyval>]{<name>}{<symbol>}{<unit>}
```

können Sie die vordefinierten Befehle überschreiben:

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

```
1 \renewstate{Enthalpy}{h
   }\joule}
2 \Enthalpy(f){12.5}
```

Der Befehl `\renewstate` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

Der Befehl verhält sich absolut analog zu `\setnewstate`, dh. er hat die gleichen Schlüssel.

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

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

```

1 \setnewstate[exponent={
    enthalpy}{h}{\kilo\
    joule\per\mole}%
    molar
2 \renewstate[exponent={
    Enthalpy}{H}{\kilo\
    joule}% absolut
3 \enthalpy{-12.3} \
    Enthalpy{-12.3}

```

12.2 State

Die in Abschnitt 12.1 vorgestellten Befehle verwenden intern alle den Befehl

`\State[<keyval>]{<symbol>}{<subscript>}`

Mit diesem Befehl können die thermodynamischen Größen auch ohne Zahl und Einheit angegeben werden.

Beispiele:

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

```

1 \State{A}, \State{G}{f},
    \State[subscript-left
    =false]{E}{\ce{Na}},
    \State[exponent=\SI
    {1000}{\celsius}]{H}

```

Wieder gibt es eine Reihe Schlüssel, mit denen das Erscheinungsbild angepasst werden kann:

- `exponent = <anything>`
- `subscript-left = true/false`
- `delta = <anything>/false`

In v1.1 gab es noch den Befehl `\setstatesubscript{<subscript pos>}`. Mit Laden der Paketooption `version=1` wird er bereitgestellt.

Beachten Sie, dass `{<subscript>}` ein *optionales* Argument ist.

Der Befehl `\State` hat seit v2.0 eine andere Syntax. Die Paketooption `version=1` stellt die ursprüngliche Syntax wieder her.

`exponent = <anything>`
 Default: `\standardstate`
 Modul: `state`

`subscript-left = true/false`
 Default: `true`
 Modul: `state`

`delta = <anything>/false`
 Default: `\Delta`
 Modul: `state`

`\setstatesubscript{<subscript pos>}` wird nicht mehr bereitgestellt. Verwenden Sie die Paketooption `version=1`, um ihn wieder zu reaktivieren.

13 Spektroskopie

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, 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`).

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

Alle Argumente dieses Befehls sind optional.

$^1\text{H-NMR: } \delta$
 $^1\text{H-NMR}$

```
1 \NMR \\  
2 \NMR*
```

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

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

```
1 \NMR{13,C}
```

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

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

```
1 \NMR(400)
```

Die verwendete Einheit kann durch explizite Angabe geändert werden:

$^1\text{H-NMR (4} \times 10^8 \text{ Hz): } \delta$

```
1 \NMR(4e8,\hertz)
```

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

$^1\text{H-NMR (4} \cdot 10^8 \text{ Hz): } \delta$

```
1 \sisetup{exponent-product  
=\cdot}\NMR(4e8,\hertz)
```

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

$^1\text{H-NMR (CDCl}_3\text{): } \delta$

```
1 \NMR[CDCl3]
```

Mit den Schlüsseln `unit` und `nucleus` können Default-Einheit und Default-Kern geändert werden.

Beispiele:

$^{13}\text{C-NMR (100 MHz): } \delta$ $^{13}\text{C-NMR (100 MHz)}$
 $^{19}\text{F-NMR (CFCl}_3\text{)}$ $^{19}\text{F-NMR (285 MHz, CFCl}_3\text{)}$
 $^1\text{H-NMR (400 MHz, CDCl}_3\text{): } \delta$
 $= 1.59$ (q, 1H, $J = 11.6$ Hz, H-4)

```
1 {\chemsetup[nmr]{nucleus  
={13,C}}\NMR(100) \NMR*(100) } \\  
2 \NMR*{19,F}[CFCl3] \NMR*{19,F}(285)[CFCl3] \\  
3 \NMR(400)[CDCl3] = \num{1.59} (q, 1H, \textit{J} = \SI{11.6}{\hertz}, H-4)
```

```
unit = <unit>  
Default: \mega\hertz  
Modul: nmr
```

```
nucleus = {<num>,<elem>}  
Default: {1,H}  
Modul: nmr
```

Wenn Sie wollen, dass das Erscheinungsbild mit dem von `bpchem` übereinstimmt (vgl. `bpchem`-Befehl `\HNMR` $^1\text{H-NMR: } \delta$ und `chemmacros`-Befehl `\NMR` $^1\text{H-NMR: } \delta$), dann verwenden Sie die Paketoption `bpchem` (siehe Abschnitt 3).

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

14 Befehle für mhchem

Ab v2.0 lädt chemmacros mhchem automatisch.

Bevor nun die Befehle 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:

Na^+ $\text{Ca}^+_{[2]}$ Ca^{2+} Ca^{2+}	<pre> 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 </pre>
--	--

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

$-OMe$ $-OMe$ $-OMe$	<pre> 1 \ce{\mch OMe}\ \ % falsche Darstellung 2 \ce{\mch{} OMe}\ \ % richtige Darstellung 3 \ce{\$\mch\$OMe} % richtige Darstellung </pre>
----------------------------	--

Das betrifft *nicht* nur chemmacros-Befehle!

<pre> 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 </pre>

$A \quad B \text{Na}_2\text{O}$ $A \quad B \text{Na}_2\text{O}$ $A \quad B \text{Na}_2\text{O}$

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

14.1 Reaktionsumgebungen

Durch **chemmacros** definiert

Es stehen die Umgebungen...

```
\begin{reaction}
<mhchem code>
\end{reaction}
\begin{reactions}
<mhchem code>
\end{reactions}
```

...sowie die Varianten...

```
\begin{reaction*}
<mhchem code>
\end{reaction*}
\begin{reactions*}
<mhchem code>
\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.

```
1 Reaktion mit Zähler:
2 \begin{reaction}
3 A -> B
4 \end{reaction}
```

Reaktion mit Zähler:



```
1 Reaktion ohne Zähler:
2 \begin{reaction*}
3 C -> D
4 \end{reaction*}
```

Reaktion ohne Zähler:



```

1 mehrere ausgerichtete Reaktionen mit Z\ahler:
2 \begin{reactions}
3 A      &-> B + C \\
4 D + E &-> F
5 \end{reactions}

```

mehrere ausgerichtete Reaktionen mit Zähler:

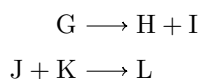


```

1 mehrere ausgerichtete Reaktionen ohne Z\ahler:
2 \begin{reactions*}
3 G      &-> H + I \\
4 J + K &-> L
5 \end{reactions*}

```

mehrere ausgerichtete Reaktionen ohne Zähler:



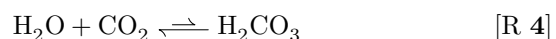
Wenn Sie das Aussehen des Zählers ändern wollen, so können Sie `\renewtagform{<tagname>}[<format>]{<right delim>}{<left delim>}`²¹ verwenden.

```

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

```

²¹ Durch das Paket `mathtools` zur Verfügung gestellt



Beachten Sie, dass sich der Name der tagform mit v2.0 geändert hat. In v1.* hieß er `CMreaction`. Die Option `version=1` stellt den alten Namen wieder her.

Eigene Reaktionen

Sie können auch neue Umgebungen selbst festlegen:

```
\newreaction[<keyval>]{<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 Schlüssel. Zum einen `star`. Damit wird zusätzlich die entsprechende Sternvariante definiert, vorausgesetzt, die gesternte Mathematik-Umgebung existiert. Wenn nicht, wird es einen Fehler geben.

Der Befehl `\newreaction` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

```

star = true/false
Default: false
kein Modul

```

Zum anderen `arg`. Damit wird eine Umgebung mit obligatorischem Argument erstellt. Auch hier gilt: die entsprechende Mathematikumgebung muss ein Argument haben.

Die bestehenden Umgebungen sind über

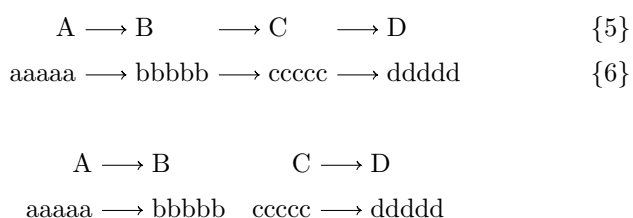
`\newreaction[star]{reaction}{equation}` und
`\reaction[star]{reactions}align`
 ...definiert.

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

`\newreaction[star,arg]{reactionsat}{alignat}`

Damit steht nun die Umgebung `reactionsat` zur Verfügung.

```
1 \newreaction[star,arg]{reactionsat}{alignat}
2 \begin{reactionsat}{3}
3 A    &-> B    &&-> C    &&-> D \\
4 aaaaa &-> bbbbb &&-> ccccc &&-> ddddd
5 \end{reactionsat}
6 \begin{reactionsat*}{2}
7 A    &-> B    & C          &-> D \\
8 aaaaa &-> bbbbb &\quad{} &-> ccccc &-> ddddd
9 \end{reactionsat*}
```



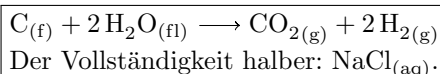
14.2 Phasen

Folgende Befehle sind für Phasenangaben gedacht. Auch wenn diese Befehle für den Einsatz mit `mhchem` gedacht waren, können sie genauso gut ohne verwendet werden.

- `\sld[<anything>]` _(f)
- `\lqd[<anything>]` _(fl)
- `\gas` _(g)
- `\aq` _(aq)

Bitte beachten Sie, dass die Befehle `\solid` und `\liquid` jetzt `\sld` bzw. `\lqd` heißen.

```
1 \ce{C\sld{} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas{}}
2 Der Vollständigkeit halber: NaCl\aq.
```

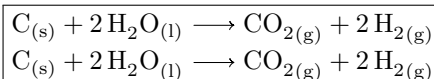


`arg = true/false`
 Default: `false`
 kein Modul

`\solid` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.
`\liquid` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

Wenn Sie lieber die englischen Bezeichnungen wollen, müssen Sie entweder auf die Paketoption `german` (siehe Abschnitt 3) verzichten, die Sprache (temporär) mit `\chemsetup[option]{german=false}` umstellen oder die Buchstaben als Argumente angeben:

```
1 {\chemsetup[option]{german=false}
2 \ce{C\sld{} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas} }\
3 \ce{C \sld[s] + 2 H2O \lqd[l] -> CO2\gas{} + 2 H2\gas}
```



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

Auch andere Einsätze sind denkbar:



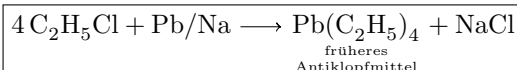
14.3 Beschriftungen

`chemmacros` stellt außerdem einen Befehl zur Verfügung, mit dem Moleküle beschriftet werden können.

```
\mhName[<keyval>]{<formula>}{<text>}
```

Zum Beispiel:

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



Mit einigen Schlüsselwörtern kann `\mhName` angepasst werden.

- `align`
- `fontattr`
- `fontsize`
- `width`

Der Befehl `\mhName` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

```
align = <alignment>
Default: \centering
Modul: mhName
```

```
fontattr = <commands>
Modul: mhName
```

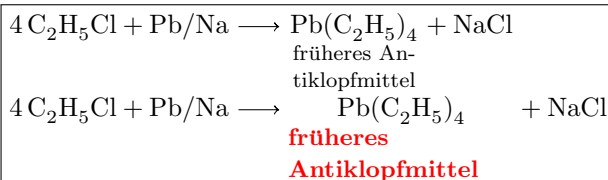
```
fontsize = <fontsize>
Default: \tiny
Modul: mhName
```

```
width = <dim>
Modul: mhName
```

```

1 \ce{4 C2H5Cl + Pb / Na -> \mhName[fontsize=\
    footnotesize]{Pb(C2H5)4}{fr\"uheres Antiklopfmittel}
    + NaCl}\
2 \chemsetup[mhName]{align=\raggedright,fontsize=\small,
    fontattr=\bfseries\color{red},width=3cm}
3 \ce{4 C2H5Cl + Pb / Na -> \mhName{Pb(C2H5)4}{fr\"uheres
    Antiklopfmittel} + NaCl}

```



In v1.1 gab es noch den Befehl `\setmhName{<textattr>}`. Mit Laden der Paketoption `version=1` wird er bereitgestellt.

`\setmhName{<textattr>}` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

15 Newman-Projektionen

chemmacros stellt das Makro

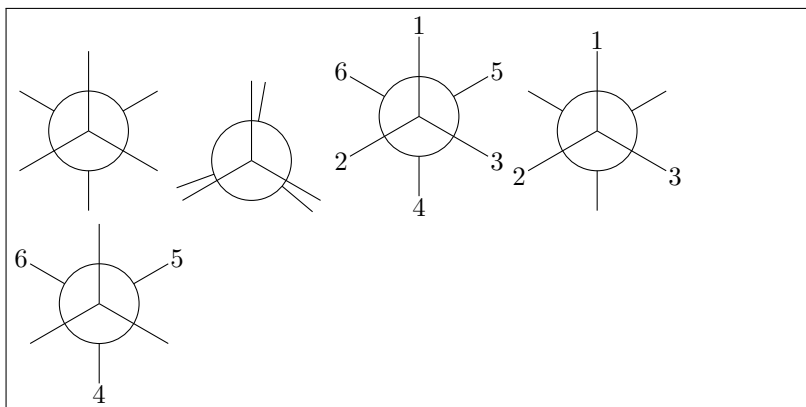
```
\newman[<keyval>](<angle>){<1>,<2>,<3>,<4>,<5>,<6>}
```

zur Verfügung, mit dem sich Newman-Projektionen erstellen lassen (verwendet ‘*TikZ*’). Mit `<angle>` lassen sich die hinteren Atome gegen die vorderen (gegen den Uhrzeigersinn) drehen.

```

1 \newman{} \newman(170){}
2 \newman{1,2,3,4,5,6} \newman{1,2,3} \newman{,,4,5,6}

```

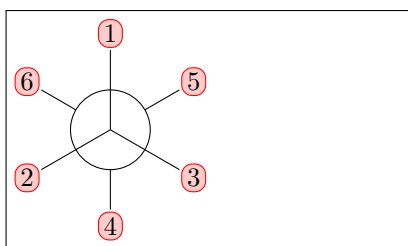


Der Befehl `\newman` hat seit v2.0 eine andere Syntax. Die Paketoption `version=1` stellt die ursprüngliche Syntax wieder her.

Mehrere Schlüssel erlauben das Anpassen des Layouts:

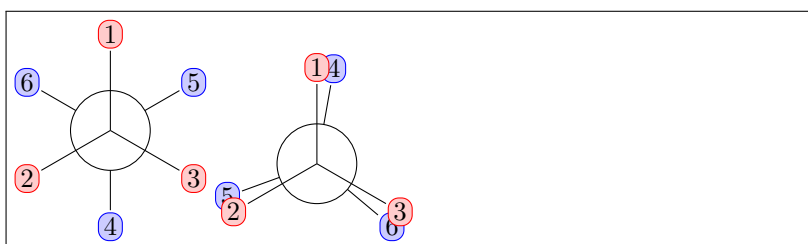
- `angle = <angle>` Einstellen des Default-Drehwinkels
- `scale = <factor>` Skalieren der Projektion
- `ring = <tikz>` Anpassen des Rings mit *TikZ*-Keys
- `atoms = <tikz>` Anpassen der Nodes, in die die Atome gesetzt werden
- `back-atoms = <tikz>` Eigene Anpassungen für die hinteren Atome

```
1 \chemsetup[newman]{angle=45} \newman{}
2 \newman[scale=.75,ring={draw=blue,fill=blue!20}] {}
```



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

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



```
angle = <angle>
Default: 0
Modul: newman
```

```
scale = <factor>
Default: 1
Modul: newman
```

```
ring = <tikz>
Modul: newman
```

```
atoms = <tikz>
Modul: newman
```

```
back-atoms = <tikz>
Modul: newman
```

16 s-, p- und Hybrid-Orbitale

In v1.1 gab es noch die Befehle `\porb[<options>]`, `\phorb[<options>]`, `\pxorb`, `\pyorb` und `\pzorb`. Mit Laden der Paketoption `version=1` werden sie bereitgestellt.

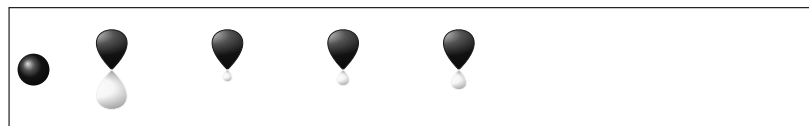
chemmacros stellt folgenden Befehl bereit, mit dem Orbitale dargestellt werden können:

`\orbital[<keyval>]{<type>}`

Als `<type>` haben Sie folgende Auswahlmöglichkeiten:

- s
- p
- sp
- sp²
- sp³

```
\orbital{s} \orbital{p} \orbital{sp} \orbital{sp2} \orbital{sp3}
```



`\porb[<options>]` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

`\phorb[<options>]` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

`\pxorb` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

`\pyorb` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

`\pzorb` wird nicht mehr bereitgestellt. Verwenden Sie die Paketoption `version=1`, um ihn wieder zu reaktivieren.

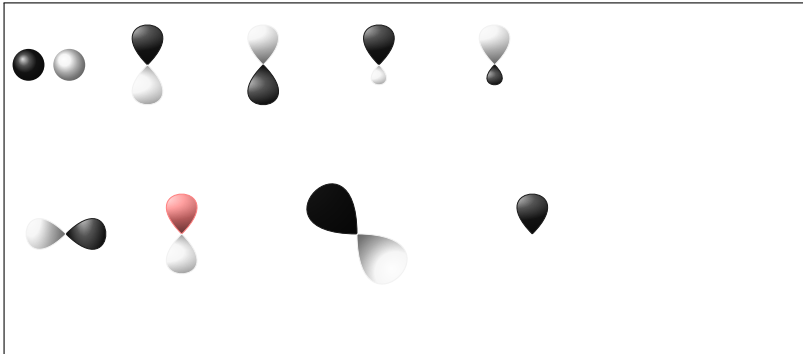
Je nach Typ gibt es verschiedene Keys zur Modifikation:

- `phase = +/-` ändert die Phase des Orbitals (alle Typen)
- `scale = <factor>` ändert die Größe des Orbitals (alle Typen)
- `color = <color>` ändert die Farbe des Orbitals (alle Typen)
- `angle = <angle>` dreht die Orbitale mit p-Anteil gegen den Uhrzeigersinn (alle Typen außer s)
- `half = true/false` stellt lediglich ein halbes Orbital dar (nur p)

```

1 \orbital{s} \orbital[phase=-]{s}
2 \orbital{p} \orbital[phase=-]{p}
3 \orbital{sp3} \orbital[phase=-]{sp3}
4
5 \orbital[angle=0]{p} \orbital[color=red!50]{p} \orbital
  [angle=135,scale=1.5]{p} \orbital[half]{p}

```



Zusätzlich gibt es zwei Keys, mit denen sich das `TikZ`-Verhalten beeinflussen lässt.

- `overlay = true/false` das Orbital „nimmt keinen Raum ein“; es wird mit der `TikZ`-Option `overlay` dargestellt.
- `opacity = <num>` das Orbital wird durchscheinend; `<value>` kann Werte von 1 (deckend) bis 0 (unsichtbar) haben.

```

phase = +/-
Default: +
Modul: orbital/<type>

```

```

scale = <factor>
Default: 1
Modul: orbital/<type>

```

```

color = <color>
Default: black
Modul: orbital/<type>

```

```

angle = <angle>
Default: 90
Modul: orbital/<type>

```

```

half = true/false
Default: false
Modul: orbital/<type>

```

```

overlay = true/false
Default: false
Modul: orbital

```

```

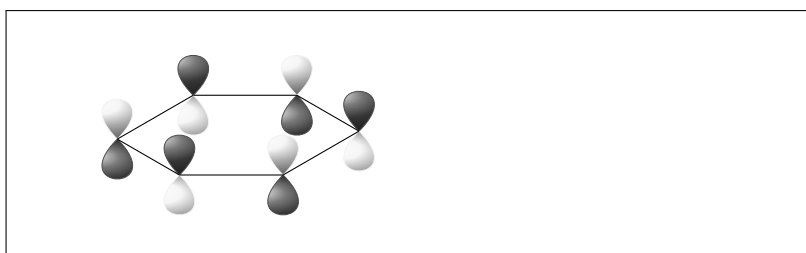
opacity = <num>
Default: 1
Modul: orbital

```

```

1 \vspace{1cm}\hspace{1cm}
2 \chemsetup[orbital]{
3   overlay,
4   p/color = black!70
5 }
6 \setbondoffset{0pt}
7 \chemfig{?\orbital{p}-[,1.3]{\orbital[phase=-]{p
   }}-[:30,1.1]\orbital{p}-[:150,.9]{\orbital[phase=-]{
   p}}-[4,1.3]\orbital{p}-[:-150,1.1]{\orbital[phase
   =-]{p}}?}
8 \vspace{1cm}

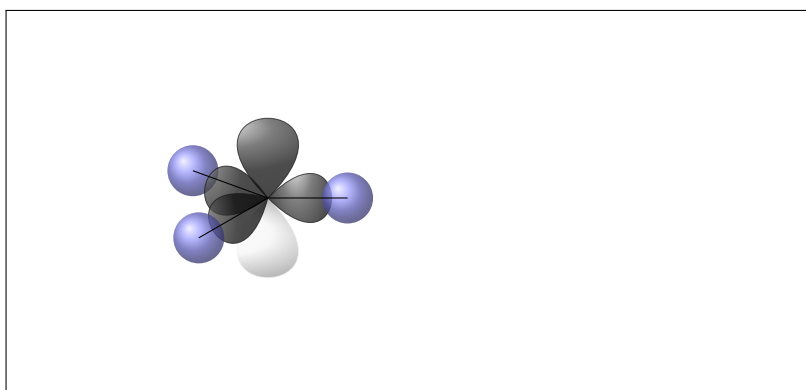
```



```

1 \vspace{2cm}\hspace{2cm}
2 \setbondoffset{0pt}
3 \chemsetup[orbital]{
4   overlay ,
5   opacity = .75 ,
6   p/scale = 1.6 ,
7   s/color = blue!50 ,
8   s/scale = 1.6
9 }
10 \chemfig{\orbital{s}-[:-20]{\orbital[scale=2]{p}}{\
   orbital[half,angle=0]{p}}{\orbital[angle=170,half]{p
   }}{\orbital[angle=-150,half]{p}}(-[:-150]\orbital{s
   })-\orbital{s}}
11 \vspace{2cm}

```



17 Schlüssel-Übersicht

In der folgenden Tabelle sind alle in chemmacros vorkommenden Schlüssel aufgelistet. Alle Schlüssel, die einem Modul angehören, können auch mit `\chemsetup[<modul>]{<keyval>}` bzw.

`\chemsetup{<modul>/<keyval>}` gesetzt werden.

Manche Schlüssel können auch ohne Wert eingesetzt werden. Sie entsprechen dann dem unterstrichenen Wert.

Schlüssel	Modul	Werte	Default	
Paket-Optionen:				
bpchem	option	<u>true</u> /false	false	Seite 3
circled	option	<u>formal</u> /all/none	formal	Seite 3
circletype	option	<u>chem</u> /math	chem	Seite 4
EZ	option	<u>chemmacros</u> /cool	chemmacros	Seite 4
german	option	<u>true</u> /false	false	Seite 4
version	option	1/2	2	Seite 4
xspace	option	<u>true</u> /false	true	Seite 4
<code>\ba</code> , <code>\Nu</code> :				
elpair	base	<u>dots</u> /dash/false	false	Seite 6
<code>\pch</code> , <code>\mch</code> , <code>\fpch</code> , <code>\fmch</code> :				
append	charges	<u>true</u> /false	false	Seite 9
<code>\ox</code> :				
parse	ox	<u>true</u> /false	true	Seite 10
roman	ox	<u>true</u> /false	true	Seite 10
pos	ox	top/super/side	top	Seite 10
explicit-sign	ox	<u>true</u> /false	false	Seite 10
decimal-marker	ox	comma/point	point	Seite 10
<code>\OX</code> , <code>\redox</code> :				
dist	redox	<dim>	.6em	Seite 13
sep	redox	<dim>	.2em	Seite 13
<code>\Enthalpy</code> , <code>\Entropy</code> , <code>\Gibbs</code> :				
exponent		<anything>	<code>\standardstate</code>	Seite 15
delta		<anything>/false		Seite 15
subscript		left/right		Seite 15
unit		<unit>		Seite 15
<code>\setnewstate</code> , <code>\renewstate</code> :				
exponent		<anything>	<code>\standardstate</code>	Seite 16
delta		<anything>/false		Seite 16
subscript		<anything>		Seite 16
subscript-left		true/false		Seite 16
<code>\State</code> :				
exponent		<anything>	<code>\standardstate</code>	Seite 17
delta		<anything>/false		Seite 17
subscript-left		true/false		Seite 17
<code>\NMR</code> :				
unit	nmr	<unit>	<code>\mega\hertz</code>	Seite 18
nucleus	nmr	{<num>,<atom symbol>}	{1,H}	Seite 18
<code>\newreaction</code> :				

Schlüssel	Modul	Werte	Default	
star		<u>true</u> /false	false	Seite 21
arg		<u>true</u> /false	false	Seite 22
\mhName:				
align	mhName	<alignment>	\centering	Seite 23
fontattr	mhName	<commands>		Seite 23
fontsize	mhName	<fontsize>	\tiny	Seite 23
width	mhName	<dim>		Seite 23
\newman:				
angle	newman	<angle>	0	Seite 25
scale	newman	<factor>	1	Seite 25
ring	newman	<tikz>		Seite 25
atoms	newman	<tikz>		Seite 25
back-atoms	newman	<tikz>		Seite 25
\orbital <type> = s/p/sp/sp2/sp3:				
phase	orbital/<type>	+/-	+	Seite 27
scale	orbital/<type>	<factor>	1	Seite 27
color	orbital/<type>	<color>	black	Seite 27
angle	orbital/<type>	<angle>	90	Seite 27
half	orbital/p	<u>true</u> /false	false	Seite 27
overlay	orbital	<u>true</u> /false	false	Seite 27
opacity	ornital	<num>	1	Seite 27

18 Vorschläge und Bugreports

Feedback über chemmacros ist jederzeit hochwillkommen! Wenn Sie Vorschläge für Makros, fehlende Funktionen usw. haben, zögern Sie nicht, Sich mit mir in Verbindung zu setzen. Wenn Ihnen Fehler auffallen, sei es fachlicher Natur, falsche Dokumentation oder dergleichen, wäre ich über eine kurze E-Mail an contact@mychemistry.eu sehr dankbar. Sollten Ihnen Bugs auffallen, senden Sie mir am besten ein Minimales Beispiel, mit dem sich der Fehler nachvollziehen lässt.

Vielen Dank an alle, die mir bereits Feedback zukommen ließen!