

CHEMFORMULA

v4.10a 2015/04/03

typeset chemical compounds and reactions

Clemens NIEDERBERGER

<https://github.com/cgnieder/chemformula/>

contact@mychemistry.eu

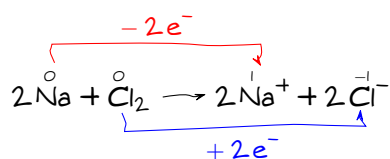


Table of Contents

1	Introduction	2	7	Compounds	9
			7.1	Adducts	9
			7.2	Subscripts	9
			7.3	Commands	10
			7.4	Charges and Other Superscripts	10
			7.5	Bonds	13
			7.5.1	Native Bonds	13
			7.5.2	Flexible Bonds	13
			7.6	Customization	15
			7.7	Standalone Formulae	19
			7.8	Extend Compound Properties	19
		8	Special Input Types	20	
			8.1	Single Token Groups	20
			8.2	Option Input	21
		9	Escaped Input	22	
			9.1	Text	22
			9.2	Math	22
		10	Arrows	23	
			10.1	Arrow types	23
2	News	2			
2.1	Version 4.0	2			
2.2	Version 4.2	2			
2.3	Version 4.3	3			
2.4	Version 4.4	3			
2.5	Version 4.5	3			
2.6	Version 4.6	3			
2.7	Version 4.7	3			
2.8	Version 4.8	3			
2.9	Version 4.9	3			
2.10	Version 4.10	4			
3	Licence and Requirements	4			
4	Setup	4			
5	The Basic Principle	4			
6	Stoichiometric Factors	6			

10.2	Labels	25	13 Usage In Math Equations	33
10.3	Customization	26		
10.4	Modify Arrow Types	27	14 Usage with TikZ or pgfplots and externalization	34
10.5	Standalone Arrows	29		
11	Names	29	15 Lewis Formulae	35
11.1	Syntax	29	16 Kröger-Vink Notation	36
11.2	Customization	30	References	37
11.3	Standalone Names	31		
12	Format and Font	31	Index	39

1 Introduction

Probably every chemist using L^AT_EX 2_ε is aware of the great mhchem package by Martin HENSEL. There have always been some difficulties intertwining it with the CHEMMACROS package, though. Also, some other minor points in mhchem always bothered me, but they hardly seemed enough for a new package. They weren't even enough for a feature request to the mhchem author. The challenge and the fun of creating a new package and the wish for a highly customizable alternative led to CHEMFORMULA after all.

CHEMFORMULA works very similar to mhchem but is more strict as to how compounds, stoichiometric factors and arrows are input. In the same time CHEMFORMULA offers *many* possibilities to customize the output.

2 News

2.1 Version 4.0

Introduced in
version 4.0

Since version 4.0, the CHEMFORMULA package can be used independently from CHEMMACROS. This means that if you say

```
1 \usepackage{chemformula}
```

then CHEMMACROS will not be loaded. The CHEMMACROS package, however, *will* load CHEMFORMULA.

2.2 Version 4.2

- New option `arrow-style`.
- New command `\chlewis` that allows to add Lewis electrons to an atom, see section 15.

2.3 Version 4.3

- New option `stoich-print`.
- New command `\chstoich`.
- The commands `\DeclareChem<...>` now don't give an error any more if the command already exists. This is more consistent with L^AT_EX's `\DeclareRobustCommand`. For all those commands a version `\NewChem<...>` is introduced that *does* give an error if the new command is already defined.

2.4 Version 4.4

- A single dash - in `\ch` is now treated as a minus sign. This is consistent with the behaviour of a +.

2.5 Version 4.5

- New arrow types `>=<`, `>=<<`, `>>=<` and `<==>`.
- Internal changes to `\ch` allow usage of optional arguments of `\` and `\label` in `CHEM-MACROS`' reactions environment.

2.6 Version 4.6

- New options `circled` and `circletype`. this allows to set the behaviour as described on `CHEMMACROS`' manual for a specific usage of `\ch`.

2.7 Version 4.7

- Dependency change: `CHEMFORMULA` now requires the TikZ library `arrows.meta` instead of the library `arrows`. This requires TikZ version 3.0.0.

2.8 Version 4.8

- The `CHEMFORMULA` package now is no longer part of the `CHEMMACROS` bundle but is distributed as a package of it's own.

2.9 Version 4.9

- New options `minus-space` and `minus-penalty` equivalent to the existing `plus-space` and `plus-penalty`
- New options `plus-output-symbol` and `minus-output-symbol` for customizing the plus and minus signs in the output.

2.10 Version 4.10

- new macros `\NewChemCompoundProperty`, `\RenewChemCompoundProperty`, `\DeclareChemCompoundProperty` and `\RemoveChemCompoundProperty`. See section 7.8 for a description.

3 Licence and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the L^AT_EX Project Public License (L^PP_L), version 1.3 or later (<http://www.latex-project.org/lppl.txt>). The software has the status “maintained.”

The `CHEMFORMULA` package needs and thus loads the packages `l3kernel` [The13a], `xparse`, `l3keys2e` and `xfrac` (all three are part of the `l3packages` bundle [The13b]), `tikz`¹ [Tan13], `ams-math` [Ameo2], `nicefrac` [Rei98] and `scrfile` (from the KOMA-Script² bundle [KN12]).

4 Setup

All of `CHEMFORMULA`’s options belong to `CHEMMACROS`’ module `chemformula`. This means if you load it via `CHEMMACROS` or in addition to `CHEMMACROS` they can be setup with

```
\chemsetup[chemformula]{<options>}
```

Set up options for `CHEMFORMULA` exclusively, or

```
\chemsetup{chemformula/<option1>,chemformula/<option2>}
```

Set up options for `CHEMFORMULA` together with others of `CHEMMACROS`’ options.

However, if you’re using `CHEMFORMULA` as a standalone package the command `\chemsetup` is not available. This is why `CHEMFORMULA` also has its own setup command:

```
\setchemformula{<options>}
```

Set up `CHEMFORMULA` when using it independently from `CHEMMACROS`.

5 The Basic Principle

`CHEMFORMULA` offers one main command.

```
\ch[<options>]{<input>}
```

`CHEMFORMULA`’s main command.

The usage will seem very familiar to you if you’re familiar with `mhchem`:

1. on CTAN as `pgf`: <http://mirrors.ctan.org/graphics/pgf/>

2. on CTAN as `koma-script`: <http://mirrors.ctan.org/macros/latex/contrib/koma-script/>

5 The Basic Principle

1 <code>\ch{H2O} \par</code>	H_2O
2 <code>\ch{Sb2O3} \par</code>	Sb_2O_3
3 <code>\ch{H+} \par</code>	H^+
4 <code>\ch{CrO4^2-} \par</code>	CrO_4^{2-}
5 <code>\ch{AgCl2-} \par</code>	AgCl_2^-
6 <code>\ch{[AgCl2]-} \par</code>	$[\text{AgCl}_2]^-$
7 <code>\ch{Y^{99+}} \par</code>	Y^{99+}
8 <code>\ch{Y^{99+}} \par</code>	Y^{99+}
9 <code>\ch{H2_{(aq)}} \par</code>	$\text{H}_{2(\text{aq})}$
10 <code>\ch{NO3-} \par</code>	NO_3^-
11 <code>\ch{(NH4)2S} \par</code>	$(\text{NH}_4)_2\text{S}$
12 <code>\ch^{227}_{90}Th+} \par</code>	${}^{227}_{90}\text{Th}^+$
13 <code>\$V_{\ch{H2O}}\$ \par</code>	$V_{\text{H}_2\text{O}}$
14 <code>\ch{Ce^{IV}} \par</code>	Ce^{IV}
15 <code>\ch{KCr(SO4)2 * 12 H2O}</code>	$\text{KCr}(\text{SO}_4)_2 \cdot 12 \text{H}_2\text{O}$

However, there are differences. The most notable one: **CHEMFORMULA** distinguishes between different types of input. These different parts *have* to be separated with blanks:

`\ch{part1 part2 part3 part4}`

A blank in the input *never* is a blank in the output. This role of the blank strictly holds and disregarding it can have unexpected results and even lead to errors.

Another notable difference: **CHEMFORMULA** tries to avoid math mode whenever possible:

1 <code>\ch{A + B ->[a] C} \par</code>	$A + B \xrightarrow{a} C$
2 <code>\ce{A + B ->[a] C}</code>	$A + B \xrightarrow{a} C$

This means that `\ch{2H2O}` is recognized as a *single* part, which in this case is recognized as a compound.

1 <code>\ch{2H2O} \par</code>	${}_2\text{H}_2\text{O}$
2 <code>\ch{2 H2O}</code>	$2 \text{H}_2\text{O}$

This also means, that a part cannot contain a blank since this will automatically divide it into two parts. If you need an extra blank in the output you need to use `~`. However, since commands in most cases gobble a space after them an input like `\ch{\command ABC}` will be treated as a single

part. If you want or need to divide them you need to add an empty group: `\ch{\command}{ABC}`. The different input types are described in the following sections.

There are some options to customize the output of the `\ch` command. They can either be applied locally using the optional argument or can be set globally using the setup command. All options of `CHEMFORMULA` belong to the module `chemformula` and can be set in different ways:

`\chemsetup[chemformula]{<options>}`

when loaded via `CHEMMACROS`

`\chemsetup{chemformula/<options>}`

when loaded via `CHEMMACROS`

`\setchemformula{<options>}`

independent from `CHEMMACROS`

6 Stoichiometric Factors

A stoichiometric factor may only contain of numbers and the signs `.`, `_`/`()`

```

1 \ch{2} \par
2 \ch{12}
3
4 % decimals:
5 \ch{.5} \par
6 \ch{5,75}
7
8 % fractions:
9 \ch{3/2} \par
10 \ch{1_1/2}
11
12 % ``iupac``:
13 \ch{(1/2)}
```

As you can see if you input decimal numbers a missing leading zero is added. You have to be a little bit careful with the right syntax but I believe it is rather intuitive.

```

1 this won't work but will result in an error: \ch{1/1_1}
```

If stoichiometric factors are enclosed with parentheses the fractions are not recognized and missing leading zeros are not added. What's inside the parentheses is typeset as is.

```
\ch{(1/2) H2O} \ch{1/2 H2O} \ch{0.5 H2O}
```

(1/2)H₂O $\frac{1}{2}$ H₂O 0.5 H₂O

You can find many examples like the following for stoichiometric factors in parentheses in the IUPAC Green Book [Coh+08]:



There are a few possibilities to customize the output.

decimal-marker = $\{\langle marker \rangle\}$ Default: .

The symbol to indicate the decimal.

frac-style = $\{\text{math}|\text{xfrac}|\text{nicefrac}\}$ Default: math

Determines how fractions are displayed.

frac-math-cmd = $\{\langle command\ sequence \rangle\}$ Default: `\frac`

Allows you to choose which command is used with **frac-style** = $\{\text{math}\}$. This needs to be a command sequence that takes two arguments that are set in math mode.

Introduced in
version 4.1

stoich-space = $\{\langle skip \rangle\}$ Default: .1667em plus .0333em minus .0117em

The space that is placed after the stoichiometric factor. A rubber length.

stoich-paren-parse = $\{\text{true}|\text{false}\}$ Default: false

If set to true stoichiometric factors enclosed by parentheses also are parsed.

stoich-print = $\{\langle cs \rangle\}$ Default: `\chstoich`

This option allows to redefine the macro that prints the stoichiometric factors. $\langle cs \rangle$ should be a macro that takes one mandatory argument. *Please note that using this option will disable **CHEMFORMULA**'s stoichiometric parsing as that is done by the default command `\chstoich`.*

```
\ch[decimal-marker={,}]{3.5} \ch[decimal-marker={\cldot}]{3,5}
```

3,5 3·5

The option **frac-style** = $\{\text{xfrac}\}$ uses the `\sfrac` command of the `xfrac` package. The output strongly depends on the font you use.

6 Stoichiometric Factors

```
1 \ch[frac-style=xfrac]{3/2} \ch[frac-style=xfrac]{1_1/2}
```

$\frac{3}{2}$ $1\frac{1}{2}$

CHEMFORMULA defines the instance `formula-text-frac` which you can redefine to your needs. See the `xfrac` documentation for further information. The default definition is this:

```
1 \DeclareInstance{xfrac}{chemformula-text-frac}{text}
2 {
3   slash-left-kern = -.15em ,
4   slash-right-kern = -.15em
5 }
```

This document uses the font Linux Libertine O and the following definition:

```
1 \DeclareInstance{xfrac}{chemformula-text-frac}{text}
2 {
3   scale-factor      = 1 ,
4   denominator-bot-sep = -.2ex ,
5   denominator-format = \scriptsize #1 ,
6   numerator-top-sep = -.2ex ,
7   numerator-format   = \scriptsize #1 ,
8   slash-right-kern   = .05em ,
9   slash-left-kern    = .05em
10 }
```

The option `frac-style = {nicefrac}` uses the `\nicefrac` command of the `nicefrac` package.

```
1 \ch[frac-style=nicefrac]{3/2} \ch[frac-style=nicefrac]{1_1/2}
```

$\frac{3}{2}$ $1\frac{1}{2}$

The option `stoich-space` allows you to customize the space between stoichiometric factor and the group following after it.

7 Compounds

1	<code>\ch{2 H2O} \par</code>	2 H ₂ O
2	<code>\ch[stoch-space=.3em]{2 H2O}</code>	2 H ₂ O

7 Compounds

CHEMFORMULA determines compounds as the type that “doesn’t fit in anywhere else.” This point will become more clear when you know what the other types are.

1	<code>\ch{H2SO4} \par</code>	H ₂ SO ₄
2	<code>\ch{[Cu(NH3)4]^2+}</code>	[Cu(NH ₃) ₄] ²⁺

7.1 Adducts

CHEMFORMULA has two identifiers which will create adducts.

`\ch{A.B}`
A·B

`\ch{A*B}`
A·B

1	<code>\ch{CaSO4.H2O} \par</code>	CaSO ₄ ·H ₂ O
2	<code>\ch{CaSO4*H2O}</code>	CaSO ₄ ·H ₂ O

Since numbers in a compound always are treated as subscripts (see section 7.2) you sometimes need to introduce stoichiometric factors for the right output:

1	<code>\ch{Na3PO4*12H2O} \par</code>	Na ₃ PO ₄ · ₁₂ H ₂ O
2	<code>\ch{Na3PO4* 12 H2O} \par</code>	Na ₃ PO ₄ ·12 H ₂ O
3	<code>\ch{Na3PO4 * 12 H2O}</code>	Na ₃ PO ₄ ·12 H ₂ O

7.2 Subscripts

All numbers in a compound are treated as subscripts.

7 Compounds

```
1 \ch{H2SO4}
```

H₂SO₄

If you want a letter to be a subscript you can use the math syntax:

```
1 \ch{A_nB_m}
```

A_nB_m

The subscript recognizes groups. You can also use math inside it.

```
1 \ch{A_{n}B_{m}} \par  
2 \ch{NaCl_{(aq)}}
```

A_nB_m
NaCl_(aq)

7.3 Commands

Commands are allowed in a compound:

```
1 \ch{\textbf{A2}B3} \ch{A2\color{red}B  
3}
```

A₂B₃ A₂B₃

However, if the commands demand numbers as argument, *e. g.*, space commands or **CHEM-MACROS'** `\ox` command the direct use will fail. This is because the numbers are treated as subscripts *before* the command expands.

```
1 \ch{A\hspace{2mm}B} will raise an error because \hspace sees something like  
2 this: \hspace{$_2$mm}. Actually not at all like it but equally bad\ldots
```

See section 9.1 for a way around this.

7.4 Charges and Other Superscripts

Basics If a compound *ends* with a plus or minus sign it will be treated as charge sign and typeset as superscript. In other places a plus is treated as a triple bond and a dash will be used as a single bond, see section 7.5.

7 Compounds

1	<code>\ch{A+B} \ch{AB+} \par</code>	$A\equiv B AB^+$
2	<code>\ch{A-B} \ch{AB-}</code>	$A-B AB^-$

For longer charge groups or other superscripts you can use the math syntax. It recognizes groups and you can use math inside them. Inside these groups neither + nor - are treated as bonds. If a dot . is inside a superscript it is treated as indicator for a radical. A * gives the excited state.

1	<code>\ch{A^{x-}} \par</code>	A^{x-}
2	<code>\ch{A^x-} \par</code>	A^{x-}
3	<code>\ch{A^{x-}} \par</code>	A^{x-}
4	<code>\ch{A^{x-\$}} \par</code>	A^{x-}
5	<code>\ch{RNO2^{-.}} \par</code>	$RNO_2^{\cdot-}$
6	<code>\ch{^31H} \par</code>	${}^3_1\text{H}$
7	<code>\ch{^146C} \par</code>	${}^{14}_6\text{C}$
8	<code>\ch{^58}_{26}Fe} \par</code>	${}^{58}_{26}\text{Fe}$
9	<code>\ch{NO^*}</code>	NO^*

Changed in
version 4.5a

Actually a dot . is not always treated as indicator for a radical: if the dot in the superscript is followed by a number it is interpreted as a decimal sign. It is typeset according to the option `decimal-marker`. This may be a good place to mention that a comma , in a superscript is also typeset according to `decimal-marker`.

1	<code>\ch{^{22,98}_{11}Na}</code>	
2	<code>\ch{^{22.98}_{11}Na}\par</code>	${}^{22.98}_{11}\text{Na}$
3	<code>\setchemformula{decimal-marker={,}}</code>	${}^{22,98}_{11}\text{Na}$
4	<code>\ch{^{22,98}_{11}Na}</code>	${}^{22,98}_{11}\text{Na}$
5	<code>\ch{^{22.98}_{11}Na}</code>	${}^{22.98}_{11}\text{Na}$

Ions and ion composites with more than one charge can be typeset quite as easy:

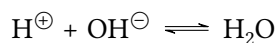
1	<code>\ch{SO4^2-} \ch{Ca^2+ SO4^2-}</code>	$SO_4^{2-} Ca^{2+}SO_4^{2-}$
---	--	------------------------------

Charge Commands You don't need to use `\mch` and related commands inside `\ch`. Indeed, you *shouldn't* use them as they might mess with the subscript and superscript alignment. The `CHEMMACROS` option circled is obeyed by `\ch`.

```

1 \chemsetup[option]{circled=all}
2 \ch{H+ + OH- <=> H2O}

```



CHEMFORMULA knows the options **circled** and **circletype** also on its own:

There are coupled with **CHEMMACROS** options, *i. e.*, setting **CHEMMACROS**' options will also set **CHEMFORMULA**'s equivalents. The other way around the options act independently: setting **CHEMFORMULA**'s options will *not* set **CHEMMACROS**' options.

circled = formal|all|none Default: formal
CHEMFORMULA uses two different kinds of charges which indicate the usage of real (+/-) and formal (\oplus/\ominus) charges. The choice **formal** distinguishes between them, choice **none** displays them all without circle, choice **all** circles all.

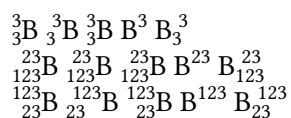
circletype = chem|math Default: chem
This option switches between two kinds of circled charge symbols: \fplus \oplus and $\text{\$oplus\$}$ \oplus .

Behaviour The superscripts behave differently depending on their position in a compound, if there are super- and subscripts following each other directly.

```

1 \ch{^33B} \ch{{}^33B} \ch{3^3B} \ch{B^3} \ch{B3^3} \par
2 \ch{^23}_{123}B \ch{{}^23}_{123}B \ch{_{123}^23}B
3 \ch{B^23} \ch{B_{123}^23} \par
4 \ch{^123}_{23}B \ch{{}^123}_{23}B \ch{_{23}^123}B
5 \ch{B^123} \ch{B23^123}

```



- If a compound *starts* with a sub- or superscript both sub- and superscript are aligned to the *right* else to the *left*.
- If a compound *does not start* with a sub- or superscript and there is both a sub- and a superscript, the superscript is shifted additionally by a length determined from the option **charge-hshift** = $\langle dim \rangle$, also see page 15f.

The second point follows IUPAC's recommendations:

TABLE 1: Bonds available with `\bond`.

name	appearance	aliases
single	—	normal, sb
double	=	db
triple	≡	tp
dotted	⋯	semisingle
deloc	≡	semidouble
tdeloc	≡	semitriple
co>	→	coordright
<co	←	coordleft

In writing the formula for a complex ion, spacing for charge number can be added (staggered arrangement), as well as parentheses: SO_4^{2-} , $(\text{SO}_4)^{2-}$. The staggered arrangement is now recommended. *IUPAC Green Book [Coh+08, p. 51]*

7.5 Bonds

7.5.1 Native Bonds

There are three kinds of what I will call “native bonds”:

1 single: <code>\ch{CH3-CH3}</code> <code>\par</code>	single: $\text{CH}_3\text{—CH}_3$
2 double: <code>\ch{CH2=CH2}</code> <code>\par</code>	double: $\text{CH}_2=\text{CH}_2$
3 triple: <code>\ch{CH+CH}</code>	triple: $\text{CH}\equiv\text{CH}$

7.5.2 Flexible Bonds

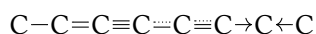
Predefined Bonds In addition to the three native bonds there are a few more which can be called by

`\bond{<bond name>}`

Prints the bond type specified by `<bond name>`.

The predefined bond types are shown in table 1.

```
1 \ch{C\bond{sb}C\bond{db}C\bond{tp}C\bond{deloc}C\bond{tdeloc}C\bond{co>}C\bond{<co}C}
```



Own Bonds `CHEMFORMULA` offers commands to define own bond types:

Introduced in
version 4.3

`\NewChemBond{<name>}{<code>}`

Define the new bond type *<name>*. Issue an error if a bond *<name>* already exists.

`\DeclareChemBond{<name>}{<code>}`

Define the new bond type *<name>* or overwrite it if it already exists.

`\RenewChemBond{<name>}{<code>}`

Redefine the existing bond type *<name>*. Issue an error if a bond *<name>* doesn't exist.

Introduced in
version 4.3

`\NewChemBondAlias{<new name>}{<old name>}`

Declare the bond type *<new name>* to be an alias of *<old name>*. Issue an error if a bond *<new name>* already exists.

`\DeclareChemBondAlias{<new name>}{<old name>}`

Declare the bond type *<new name>* to be an alias of *<old name>*.

`\ShowChemBond{<name>}`

Print the definition of bond type *<name>*.

The usage is best described with an example. So let's see how the single bond and the `co` bond are defined:

```

1 \NewChemBond{single}
2 { \draw[chembond] (chemformula-bond-start) -- (chemformula-bond-end) ; }
3 \NewChemBond{coordright}
4 {
5   \draw[chembond,butt cap->]
6     (chemformula-bond-start) -- (chemformula-bond-end) ;
7 }
8 \NewChemBondAlias{co>}{coordright}

```

Two points are important: the names of the starting and the ending coordinates, `chemformula-bond-start` and `chemformula-bond-end`, and the TikZ style of the bonds `chembond`.

So, let's say you want to define a special kind of dashed bond. You could do this:

```

1 \usetikzlibrary{decorations.pathreplacing}
2 \makeatletter
3 \NewChemBond{dashed}
4 {

```

```

5   \draw[
6     chembond,
7     decorate,
8     decoration={
9       ticks,
10      segment length=\chemformula@bondlength/10,amplitude=1.5pt
11    }]
12    (chemformula-bond-start) -- (chemformula-bond-end) ;
13  }
14  \makeatother
15  \chemsetup[chemformula]{bond-length=2ex}
16  \ch{C\bond{dashed}C}

```



The last example showed you another macro: `\chemformula@bondlength`. It only exists so you can use it to access the bond length as set with `bond-length` directly.

7.6 Customization

These options allow you to customize the output of the compounds:

`subscript-vshift = {<dim>}` Default: 0pt
 Extra vertical shift of the subscripts.

`subscript-style = text|math` Default: text
 Style that is used to typeset the subscripts.

`charge-hshift = {<dim>}` Default: .25em
 Shift of superscripts when following a subscript.

`charge-vshift = {<dim>}` Default: 0pt
 Extra vertical shift of the superscripts.

`charge-style = text|math` Default: text
 Style that is used to typeset the superscripts.

`circled = formal|all|none` Default: formal
 Like `CHEMMACROS`' package option but local to `CHEMFORMULA`'s `\ch`. That is: since `CHEMMACROS`' macros use `CHEMFORMULA`'s mechanism this is effectively an alias.

Introduced in
version 4.6

`circletype = chem|math` Default: chem
 Like `CHEMMACROS`' package option but local to `CHEMFORMULA`'s `\ch`. That is: since `CHEMMACROS`' macros use `CHEMFORMULA`'s mechanism this is effectively an alias.

Introduced in
version 4.6

`adduct-space = {<dim>}` Default: .1333em
 Space to the left and the right of the adduct point.

7 Compounds

`bond-length = {⟨dim⟩}` Default: .5833em
The length of the bonds.

`bond-offset = {⟨dim⟩}` Default: .07em
Space between bond and atoms.

`bond-style = {⟨TikZ⟩}` (initially empty)
TikZ options for the bonds.

`bond-penalty = {⟨num⟩}` Default: 10000
The penalty that is inserted after a bond for (dis-)allowing line breaks.

Introduced in
version 4.0a

`radical-style = {⟨TikZ⟩}` (initially empty)
TikZ options for the radical point.

`radical-radius = {⟨dim⟩}` Default: .2ex
The radius of the radical point.

`radical-hshift = {⟨dim⟩}` Default: .15em
Horizontal shift before the radical point is drawn.

`radical-vshift = {⟨dim⟩}` Default: .5ex
Vertical shift relative to the current baseline.

`radical-space = {⟨dim⟩}` Default: .15em
Horizontal shift after the radical point is drawn.

Maybe you have noticed that charges of certain ions are shifted to the right.

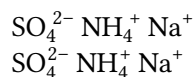
<code>1 \ch{SO4^2-} \ch{NH4+} \ch{Na+}</code>	$\text{SO}_4^{2-} \text{NH}_4^+ \text{Na}^+$
---	--

They are shifted if they *follow* a subscript which follows IUPAC recommendations [Coh+08, p. 51]. The amount of the shift can be set with the option `charge-hshift`.

<code>1 \ch{SO4^2-} \ch{NH4+} \ch{Na+} \par</code> <code>2 \chemsetup[chemformula]{charge-hshift=.5ex}</code> <code>3 \ch{SO4^2-} \ch{NH4+} \ch{Na+} \par</code> <code>4 \chemsetup[chemformula]{charge-hshift=.5pt}</code> <code>5 \ch{SO4^2-} \ch{NH4+} \ch{Na+}</code>

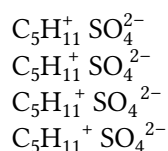
$\text{SO}_4^{2-} \text{NH}_4^+ \text{Na}^+$

7 Compounds



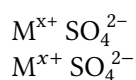
Despite IUPAC's recommendation `CHEMFORMULA` does not make fully staggered arrangements in the default setting as I find it hard to read in some cases and ugly in others. Since this is a subjective decision `CHEMFORMULA` not only let's you define the absolute amount of the shift but also provides a possibility for full staggered arrangements. For this you have to use `charge-hshift = {full}`.

```
1 \ch[charge-hshift=0pt]{C5H11+} \ch[charge-hshift=0pt]{S04^2-} \par
2 \ch{C5H11+} \ch{S04^2-} \par
3 \ch[charge-hshift=1ex]{C5H11+} \ch[charge-hshift=1ex]{S04^2-} \par
4 \ch[charge-hshift=full]{C5H11+} \ch[charge-hshift=full]{S04^2-}
```



If you don't want the charges to be typeset in text mode you can switch to math mode:

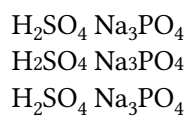
```
1 \ch{M^x+} \ch{S04^2-} \par
2 \chemsetup[chemformula]{charge-style = math}
3 \ch{M^x+} \ch{S04^2-}
```



The option `subscript-vshift` can be used to adjust the vertical shift of the subscripts:

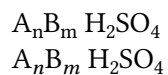
```
1 \ch{H2S04} \ch{Na3P04} \par
2 \chemsetup[chemformula]{subscript-vshift=.5ex}
3 \ch{H2S04} \ch{Na3P04} \par
4 \chemsetup[chemformula]{subscript-vshift=-.2ex}
5 \ch{H2S04} \ch{Na3P04}
```

7 Compounds



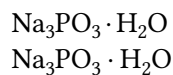
You can choose the mode subscripts are typeset in the same way as it is possible for the charges:

```
1 \ch{A_nB_m} \ch{H2SO4} \par
2 \chemsetup[chemformula]{subscript-style = math}
3 \ch{A_nB_m} \ch{H2SO4}
```



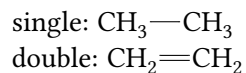
The option `adduct-space` sets the space left and right to the adduct symbol .

```
1 \ch{Na3PO3*H2O} \par
2 \chemsetup[chemformula]{adduct-space=.2em}
3 \ch{Na3PO3*H2O}
```



Changing the length of the bonds:

```
1 \chemsetup[chemformula]{bond-length=4mm}%
2 single: \ch{CH3-CH3} \par
3 double: \ch{CH2=CH2} \par
4 triple: \ch{CH+CH}
```



triple: CH≡CH

You can change the distance between bond and atom, too:

```
1 \ch{H-H + N+N + O=O} \par
2 \ch[bond-offset=1pt]{H-H + N+N + O=O}
```

H–H + N≡N + O=O

H–H + N≡N + O=O

7.7 Standalone Formulae

Introduced in
version 4.0

CHEMFORMULA offers a command that *only accepts* the “compound” input type:

`\chcpd[⟨options⟩]{⟨compound⟩}`

Typeset single compounds.

7.8 Extend Compound Properties

Introduced in
version 4.10

It is possible to extend the range of special input symbols within compounds. In the default setting those are `*`, `-`, `=`, `+` and arabic numerals. Others can be added or the existing ones be changed with one of the following commands:

`\NewChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`

`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property is only added if `⟨token⟩` is not yet part the compounds’ property list.

`\RenewChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`

`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property is only added if `⟨token⟩` is already part the compounds’ property list.

`\DeclareChemCompoundProperty{⟨token⟩}{⟨replacement⟩}`

`⟨token⟩` will be replaced by `⟨replacement⟩` within compounds. The property silently overwrites any previously set `⟨replacement⟩` for `⟨token⟩` if `⟨token⟩` is already part the compounds’ property list.

`\RemoveChemCompoundProperty{⟨token⟩}`

Removes `⟨token⟩` from the compounds’ property list.

For example you can use

```
\NewChemCompoundProperty{\}{\slash}
```

to allow line breaks after slashes in compounds.

8 Special Input Types

There are some “special type” input groups.

8.1 Single Token Groups

The first kind are groups which consist of only one token, namely of the following ones:

`\ch{ + } +`

Creates the plus sign between compounds with space around it:

`\ch{2 Na + Cl2}` 2 Na + Cl₂

`\ch{ - } -`

Creates the minus sign between compounds with space around it:

`\ch{M - H}` M – H

`\ch{ v } ↓`

Sign for precipitate: `\ch{BaSO4 v}` BaSO₄↓

`\ch{ ^ } ↑`

Sign for escaping gas³: `\ch{H2 ^}` H₂↑

The space left and right of the plus and the minus sign and the signs themselves can be set with the following options:

`plus-space = {<skip>}`

A rubber length.

Default: .3em plus .1em minus .1em

`minus-space = {<skip>}`

A rubber length.

Default: .3em plus .1em minus .1em

`plus-penalty = {<num>}`

The penalty that is inserted after the plus sign for (dis-)allowing line breaks.

Default: 700

`minus-penalty = {<num>}`

The penalty that is inserted after the minus sign for (dis-)allowing line breaks.

Default: 700

`plus-output-symbol = {<code>}`

The `<code>` that is used for the plus sign.

Default: +

³. Is this the correct English term? Please correct me if it isn't.

Introduced in
version 4.3a

Introduced in
version 4.9

Introduced in
version 4.0a

Introduced in
version 4.9

Introduced in
version 4.9

Introduced in
version 4.9

`minus-output-symbol = {<code>}`

Default: \$-\$

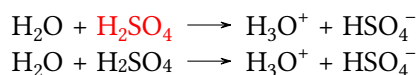
The `<code>` that is used for the minus sign.

1	<code>\ch{A + B}\par</code>	A + B
2	<code>\ch[plus-space=4pt]{A + B}</code>	A + B

8.2 Option Input

Sometimes you might want to apply an option only to a part of a, say, reaction. Of course you have the possibility to use `\ch` several times.

1	<code>\ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HS04-} \par</code>
2	<code>\ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HS04-}</code>

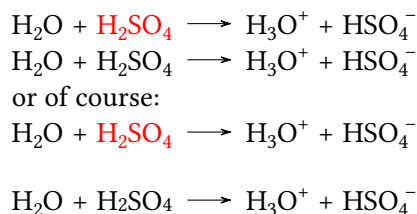


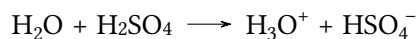
This, however, interrupts the input in your source and *may* mess with the spacing. That's why there is an alternative:

`\ch{ @<options> }`

The options specified this way will be valid *only* until the next compound is set.

1	<code>\ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HS04-} \par</code>
2	<code>\ch{H2O + @format=\color{red}} H2SO4 -> H3O+ + HS04-} \par</code>
3	or of course:\par
4	<code>\ch{H2O + \textcolor{red}{H2SO4} -> H3O+ + HS04-}\par\bigskip</code>
5	<code>\ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HS04-} \par</code>
6	<code>\ch{H2O + @subscript-vshift=2pt} H2SO4 -> H3O+ + HS04-}</code>





9 Escaped Input

In some cases it may be desirable to prevent `CHEMFORMULA` from parsing the input. This can be done in two ways.

9.1 Text

If you put something between " " or ' ' then the input will be treated as normal text, except that spaces are not allowed and have to be input with `~`.

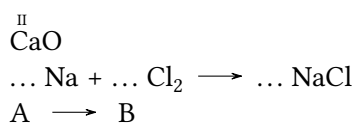
```
\ch{ "<escaped text>" }
```

One of two possibilities to *escape* `CHEMFORMULA`'s parsing.

```
\ch{ '<escaped text>' }
```

The second of two possibilities to *escape* `CHEMFORMULA`'s parsing.

```
1 \ch{"\ox{2,Ca}" 0} \par
2 \ch{"\ldots\," Na + "\ldots\," Cl2 -> "\ldots\," NaCl} \par
3 \ch{'A~->~B'}
```



In many cases you won't need to escape the input. But when you get into trouble when using a command inside `\ch` try hiding it.

9.2 Math

If you especially want to input math you just enclose it with `$ $`. This output is different from the escaped text as it is followed by a space. The reasoning behind this is that I assume math will mostly be used to replace stoichiometric factors.

```
\ch{ $<escaped math>$ }
```

One of two possibilities to *escape* `CHEMFORMULA`'s parsing into math mode.

```
\ch{ \(<escaped math>\) }
```

The second of two possibilities to *escape* `CHEMFORMULA`'s parsing into math mode.

1 escaped text: <code>\ch{"\$x\$" H2O} \par</code>	escaped text: $x\text{H}_2\text{O}$
2 escaped math: <code>\ch{\$x\$ H2O} \par</code>	escaped math: $x\text{H}_2\text{O}$
3 also escaped math: <code>\ch{\(x\) H2O} \par</code>	also escaped math: $x\text{H}_2\text{O}$
4 <code>\ch{\$2n\$ Na + \$n\$ Cl2 -> \$2n\$ NaCl}</code>	$2n\text{Na} + n\text{Cl}_2 \longrightarrow 2n\text{NaCl}$

The space that is inserted after a math group can be edited:

`math-space = {\langle skip\rangle}`

Default: .1667em plus .0333em minus .0117em

A rubber length.

```
1 \ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
2 \chemsetup[chemformula]{math-space=.25em}
3 \ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
4 \ch{$A->B$}
```

$2n\text{Na} + n\text{Cl}_2 \longrightarrow 2n\text{NaCl}$
 $2n\text{Na} + n\text{Cl}_2 \longrightarrow 2n\text{NaCl}$
 $A \rightarrow B$

10 Arrows

10.1 Arrow types

Arrows are input in the same intuitive way they are with mhchem. There are various different types:

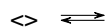
`->` \longrightarrow
 standard right arrow

`<-` \longleftarrow
 standard left arrow

`-/>` \nrightarrow
 does not react (right)

`</-` \nleftarrow
 does not react (left)

`<->` \longleftrightarrow
 resonance arrow



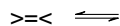
reaction in both directions



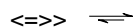
stoichiometric equation



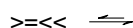
equilibrium arrow



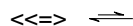
reversed equilibrium arrow



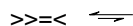
unbalanced equilibrium arrow to the right



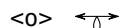
reversed unbalanced equilibrium arrow to the right



unbalanced equilibrium arrow to the left



reversed unbalanced equilibrium arrow to the left



isolobal arrow



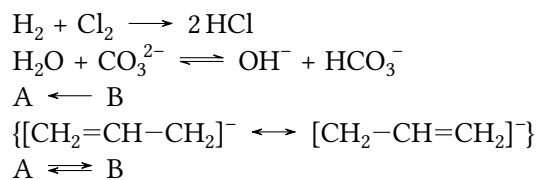
I've seen this one used. I'm not sure it actually has a meaning in chemical equations. If you have some official reference for this arrow type please feel free to contact me.

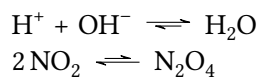
All these arrows are drawn with TikZ.

```

1 \ch{H2 + Cl2 -> 2 HCl} \par
2 \ch{H2O + CO3^2- <=> OH- + HCO3-} \par
3 \ch{A <- B} \par
4 \ch{\{[CH2=CH-CH2]- <-> \}[CH2-CH=CH2]- \} \par
5 \ch{A <> B} \par
6 \ch{H+ + OH- <=>> H2O} \par
7 \ch{2 NO2 <<=> N2O4}

```



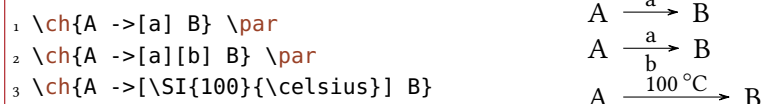


10.2 Labels

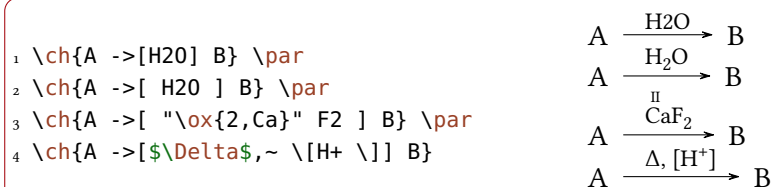
The arrows take two optional arguments to label them.

`->[<above>][<below>]`

Add text above or under an arrow.



The label text can be parsed separately from the arrow. The recipe is easy: leave blanks.



If you leave the blanks `CHEMFORMULA` treats the groups inside the square brackets as separated input types. The arrow reads its arguments *afterwards*. As you can see the arrows “grow” with the length of the labels. What stays constant is the part that protrudes the labels. As you also can see in the last example square brackets inside the arrow arguments can be produced using `\[` and `\]`. They keep their usual meaning outside `\ch`. These commands were necessary since the usual grouping (*i. e.*, hiding the brackets inside curly brackets) didn’t work due to the way `\ch` read its argument. This is no longer true but meanwhile `\[` and `\]` are kept for backwards compatibility.

```
1 \ch{A ->[a] B} \par
2 \ch{A ->[ab] B} \par
3 \ch{A ->[abc] B} \par
4 \ch{A ->[abc~abc] B} \par
5 % needs the 'chemfig' package:
6 \setatomsep{15pt}
7 \ch{A ->[ "\chemfig{-[:30]-[:-30]OH}" ] B} \par
8 \ch{A ->[[ ]] B} vs. \ch{A ->[\[ \] ] B}
```



10.3 Customization

These are the options which enable you to customize the arrows:

`arrow-offset = {<dim>}` Default: .75em

This is the length that an arrow protrudes a label on both sides. This means an empty arrow's length is two times `arrow-offset`.

`arrow-min-length = {<dim>}` Default: 0pt

Introduced in
version 3.6b

The minimal length an error must have unless two times `arrow-offset = {<p>}` plus the width of the label is larger.

`arrow-yshift = {<dim>}` Default: 0pt

Shifts an arrow up (positive value) or down (negative value).

`arrow-ratio = {<factor>}` Default: .6

The ratio of the arrow lengths of the unbalanced equilibrium. .4 would mean that the length of the shorter arrow is 0.4× the length of the longer arrow.

`compound-sep = {<dim>}` Default: .5em

The space between compounds and the arrows.

`label-offset = {<dim>}` Default: 2pt

The space between the labels and the arrows.

`label-style = {}` Default: `\footnotesize`

The relative font size of the labels.

`arrow-penalty = {<num>}` Default: 0

Introduced in
version 4.0a

The penalty that is inserted after an arrow for (dis-)allowing line breaks.

`arrow-style = {<TikZ>}` (initially empty)

Introduced in
version 4.1a

Additional TikZ keys for formatting the arrows.

The following code shows the effect of the different options on the `<=>` arrow:

```

1 standard: \ch{A <=>>[x][y] B} \par
2 longer: \ch[arrow-offset=12pt]{A <=>>[x][y] B} \par
3 higher: \ch[arrow-yshift=2pt]{A <=>>[x][y] B} \par
4 more balanced: \ch[arrow-ratio=.8]{A <=>>[x][y] B} \par
5 labels further away: \ch[label-offset=4pt]{A <=>>[x][y] B} \par
6 larger distance to compounds: \ch[compound-sep=2ex]{A <=>>[x][y] B} \par
7 smaller labels: \ch[label-style=\tiny]{A <=>>[x][y] B}

```

standard: A $\xrightarrow[y]{x}$ B
longer: A $\xrightarrow[y]{x}$ B
higher: A $\xrightarrow[y]{x}$ B
more balanced: A $\xrightarrow[y]{x}$ B
labels further away: A $\xrightarrow[y]{x}$ B
larger distance to compounds: A $\xrightarrow[y]{x}$ B
smaller labels: A $\xrightarrow[y]{x}$ B

Introduced in
version 4.7

If you want to have different arrow tips there is an easy way to use existing arrow tips (as defined by TikZ). **CHEMFORMULA** uses three different arrow tips: cf, left cf and right cf. If you want them to match those of chemfig [Tel13] for example you could do:

```

1 \pgfkeys{
2   cf /.tip = {CF@full} ,
3   left cf /.tip = {CF@half}
4 }

```

chemfig has no equivalent of right cf. This mechanism relies on TikZ version 3.0.0 and the new arrows.meta library.

10.4 Modify Arrow Types

The arrows are defined with the commands

\NewChemArrow{*<type>*}{*<TikZ>*}

Define the new arrow type *<type>*. Issue an error if an arrow type *<type>* already exists.

\DeclareChemArrow{*<type>*}{*<TikZ>*}

Define the new arrow type *<type>* or overwrite it if it already exists.

\RenewChemArrow{*<type>*}{*<TikZ>*}

Redefine the arrow type *<type>*. Issue an error if an arrow type *<type>* doesn't exist.

`\ShowChemArrow{<type>}`

Print out the current definition of the arrow type *<type>*.

<type> is the sequence of tokens that is replaced with the actual arrow code. For example the basic arrow is defined via

```

1 \NewChemArrow{->}{
2   \draw[chemarrow, -cf] (cf_arrow_start) -- (cf_arrow_end) ;
3 }

```

In order to define arrows yourself you need to know the basics of TikZ.⁴ The predefined arrows use the arrow tips `cf`, `left cf` and `right cf`. They also all except the net reaction arrow `==` use the TikZ-style `chemarrow` that you should use, too, if you want the option `arrow-style` to have an effect.

There are some predefined coordinates you can and should use. For completeness' sake the arrow tips and the TikZ-style are also listed:

`(cf_arrow_start)`

The beginning of the arrow.

`(cf_arrow_end)`

The end of the arrow.

`(cf_arrow_mid)`

The mid of the arrow.

`(cf_arrow_mid_start)`

The beginning of the shorter arrow in types like `<=>`.

`(cf_arrow_mid_end)`

The end of the shorter arrow in types like `<=>`.

`cf`

A double-sided arrow tip.

`left cf`

A left-sided arrow tip.

`right cf`

A right-sided arrow tip.

`chemarrow`

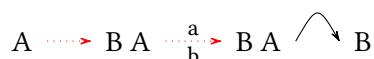
`CHEMFORMULA`'s TikZ-style that is applied to the arrows and set with `arrow-style`

4. Please see the pgfmanual for details.

```

1 \NewChemArrow{.>}{
2   \draw[chemarrow,-cf,dotted,red] (cf_arrow_start) -- (cf_arrow_end);
3 }
4 \NewChemArrow{n>}{
5   \draw[chemarrow,-cf]
6     (cf_arrow_start)
7     .. controls ([yshift=3ex]cf_arrow_mid) ..
8     (cf_arrow_end);
9 }
10 \ch{A .> B} \ch{A .>[a][b] B} \ch{A n> B}

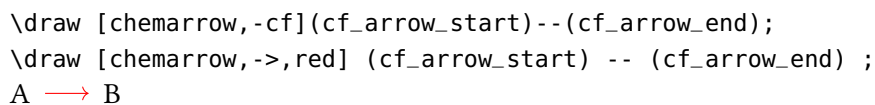
```



```

1 \texttt{\ShowChemArrow{->}} \par
2 \RenewChemArrow{->}{\draw[chemarrow,->,red] (cf_arrow_start) -- (cf_arrow_end)
3   ;}
4 \texttt{\ShowChemArrow{->}} \par
5 \ch{A -> B}

```



10.5 Standalone Arrows

Introduced in
version 4.0

CHEMFORMULA offers a command that *only accepts* the “arrow” input type:

```
\charrow{<type>}[<above>][<below>]
```

Print the arrow type *<type>*.

This command is internally used for the arrows, too, when `\ch` is parsed.

11 Names

11.1 Syntax

CHEMFORMULA has a built-in syntax to write text under a compound. In a way it works very similar to the arrows.

`\ch{!(<text>)(<formula>) }`
Writes *<text>* below *<formula>*.

If an exclamation mark is followed by a pair of parentheses **CHEMFORMULA** will parse it this way:

```
1 \ch{!(ethanol)( CH2CH2OH )}
```

CH₂CH₂OH
ethanol

The same what's true for the arrows arguments holds for these arguments: if you leave blanks the different parts will be treated according to their input type before the text is set below the formula.

```
1 \ch{!(water)(H2O)} \quad
2 \ch{!( "\textcolor{blue}{water}" )( H2O )} \quad
3 \ch{!( $2n-1$ )( H2O )} \quad
4 \ch{!( H2O )( H2O )} \quad
5 \ch{!(oxonium)( H3O+ )}
```

H ₂ O	H ₂ O	H ₂ O	H ₂ O	H ₃ O ⁺
water	water	2n - 1	H ₂ O	oxonium

If for some reason you want to insert an exclamation mark *without* it creating a name you only have to make sure it isn't followed by parentheses.

```
1 \ch{H2O~(!)} \par           H2O (!)
2 \ch{A!{ }()}                A!()
```

11.2 Customization

CHEMFORMULA provides two options to customize the output of the names:

`name-format = {<commands>}`

Default: `\scriptsize\centering`

The format of the name. This can be arbitrary input.

`name-width = <dim>|auto`

Default: auto

The width of the box where the label is put into. auto will detect the width of the name and set the box to this width.

```
1 \ch{!(acid)( H2SO4 ) -> B} \par
2 \ch[name-format=\sffamily\small]{!(acid)( H2SO4 ) -> B} \par
3 \ch[name-format=\scriptsize N:~]{!(acid)( H2SO4 ) -> B} \par
4 \ch[name-width=3em,name-format=\scriptsize\raggedright]{!(acid)( H2SO4 ) -> B}
```

H₂SO₄ → B
 acid
 H₂SO₄ → B
 acid
 H₂SO₄ → B
 N: acid
 H₂SO₄ → B
 acid

11.3 Standalone Names

Introduced in
version 4.0

CHEMFORMULA offers a command that allows the usage of the “name” syntax in normal text. This is the command that a bang is replaced with in **CHEMFORMULA**’s formulas, actually. Both arguments are mandatory.

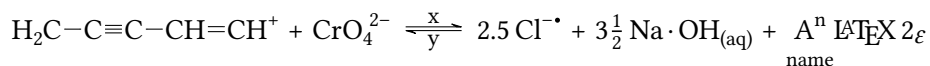
`\chname(<text 1>)(<text 2>)`

The command that is used internally for placing *<text 1>* below of *<text 2>*.

12 Format and Font

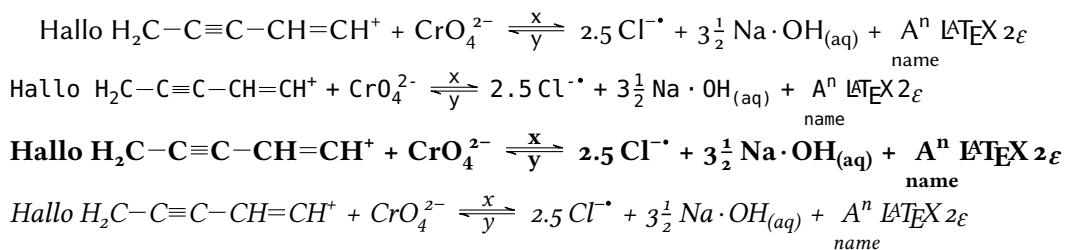
In the standard setting **CHEMFORMULA** doesn’t make any default changes to the font of the formula output. Let’s take a look at a nonsense input which shows all features:

```
1 \newcommand*\sample{%
2   \ch{H2C-C+C-CH=CH+ + CrO4^2-
3     <=>[x][y]
4     2.5 Cl^{-.} + 3_1/2 Na*OH_{(aq)} + !(name)( A^n ) "\LaTeXe"}
5 }
6 \sample
```



Now we're going to change different aspects of the font a look what happens:

```
1 \sffamily Hallo \sample \\
2 \ttfamily Hallo \sample \normalfont \\
3 \bfseries Hallo \sample \normalfont \\
4 \itshape Hallo \sample
```



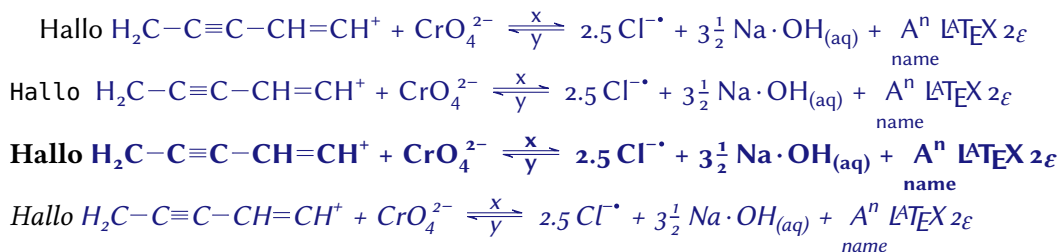
As you can see most features adapt to the surrounding font.

If you want to change the default format you need to use this option:

`format = {<anything>}` (initially empty)

Adds arbitrary code before the output of `\ch`.

```
1 \definecolor{newblue}{rgb}{.1,.1,.5}
2 \chemsetup[chemformula]{format=\color{newblue}\sffamily}
3 \sffamily Hallo \sample \\
4 \ttfamily Hallo \sample \normalfont \\
5 \bfseries Hallo \sample \normalfont \\
6 \itshape Hallo \sample
```



You can also specifically change the fontfamily, fontseries and fontshape of the output.

`font-family = {⟨family⟩}` (initially empty)

Changes the fontfamily of the output with `\fontfamily{⟨family⟩}`.

`font-series = {⟨series⟩}` (initially empty)

Changes the fontseries of the output with `\fontseries{⟨series⟩}`.

`font-shape = {⟨shape⟩}` (initially empty)

Changes the fontshape of the output with `\fontshape{⟨shape⟩}`.

```

1 \chemsetup[chemformula]{font-series=bx}
2 Hallo \sample \par
3 \sffamily Hallo \sample \normalfont \par
4 \chemsetup[chemformula]{font-family=lmss,font-series=m} Hallo \sample
5 \normalfont \par
6 \itshape Hallo \sample

```

Hallo $\text{H}_2\text{C}-\text{C}\equiv\text{C}-\text{CH}=\text{CH}^+ + \text{CrO}_4^{2-} \xrightleftharpoons[\text{y}]{\text{x}} 2.5 \text{Cl}^- + 3\frac{1}{2} \text{Na} \cdot \text{OH}_{(\text{aq})} + \text{A}^{\text{n}} \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\text{E}}$
name

Hallo $\text{H}_2\text{C}-\text{C}\equiv\text{C}-\text{CH}=\text{CH}^+ + \text{CrO}_4^{2-} \xrightleftharpoons[\text{y}]{\text{x}} 2.5 \text{Cl}^- + 3\frac{1}{2} \text{Na} \cdot \text{OH}_{(\text{aq})} + \text{A}^{\text{n}} \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\text{E}}$
name

Hallo $\text{H}_2\text{C}-\text{C}\equiv\text{C}-\text{CH}=\text{CH}^+ + \text{CrO}_4^{2-} \xrightleftharpoons[\text{y}]{\text{x}} 2.5 \text{Cl}^- + 3\frac{1}{2} \text{Na} \cdot \text{OH}_{(\text{aq})} + \text{A}^{\text{n}} \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\text{E}}$
name

Hallo $\text{H}_2\text{C}-\text{C}\equiv\text{C}-\text{CH}=\text{CH}^+ + \text{CrO}_4^{2-} \xrightleftharpoons[\text{y}]{\text{x}} 2.5 \text{Cl}^- + 3\frac{1}{2} \text{Na} \cdot \text{OH}_{(\text{aq})} + \text{A}^{\text{n}} \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\text{E}}$
name

If you're using $\text{X}_{\text{Y}}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ or $\text{L}^{\text{u}}\text{a}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ and have loaded fontspec you have the possibility to set the font with it:

`font-spec = {⟨font⟩}` (initially empty)

Use font `⟨font⟩` for `CHEMFORMULA`'s formulas.

or with options

`font-spec = {[⟨options⟩]}⟨font⟩`

Use font `⟨font⟩` with options `⟨options⟩` for `CHEMFORMULA`'s formulas.

Since this document is typeset with $\text{p}^{\text{d}}\text{fL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ the option cannot be demonstrated here.

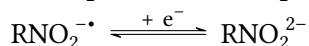
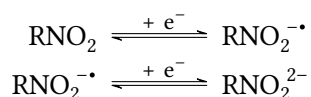
13 Usage In Math Equations

The `\ch` command can be used inside math equations. It recognizes `\` and `&` and passes them on. However, you can't use the optional arguments of `\` inside `\ch`.

```

1 \begin{align}
2 \ch{
3   H2O & \xrightarrow{a} H2SO4 \\
4   Cl2 & \xrightarrow{x} CH4
5 }
6 \end{align}
7 \begin{align*}
8 \ch{
9   RNO2 & \xrightleftharpoons{+ e^-} RNO2^{\cdot-} \\
10  RNO2^{\cdot-} & \xrightleftharpoons{+ e^-} RNO2^{2-}
11 }
12 \end{align*}

```



14 Usage with TikZ or pgfplots and externalization

Introduced in
version 4.1

Since `CHEMFORMULA` uses TikZ to draw reaction arrows and bonds they would be externalized, too, if you use that facility with TikZ or pgfplots⁵ [Feu13]. This may not be desirable since they are very small pictures maybe containing of a single line. This is why `CHEMFORMULA`'s default behaviour is to disable externalization for it's bonds and arrows. This can be turned on and off through the following option:

`tikz-external-disable = true|false` Default: true
dis- or enable TikZ' externalization mechanism for `CHEMFORMULA`'s arrows and bonds.

If you should be using a formula that contains bonds or arrows inside of a tikzpicture that is externalized you should locally enable it for `CHEMFORMULA`, too:

```

1 \begin{tikzpicture}
2   \setchemformula{tikz-external-disable=false}
3   \begin{axis}[xlabel={\ch{2 H+ + 2 e- -> H2}}]
4     \addplot ... ;

```

5. on CTAN as pgfplots: <http://mirrors.ctan.org/macros/latex/contrib/pgfplots/>

```

5 \end{axis}
6 \end{tikzpicture}

```

15 Lewis Formulae

Introduced in
version 4.2

CHEMFORMULA offers a command to typeset Lewis formulae. This does not mean Lewis structures! Those can be achieved using the chemfig package [Tel13]. **CHEMFORMULA** provides the possibility to draw electrons as dots and pairs of dots or a line around an atom.

`\chlewis`[*<options>*]{*<electron spec>*}{*<atom>*}

Draws electrons around the *<atom>* according to *<electron spec>*.

Electrons are specified by the angle to the horizontal in the counter-clockwise direction. The default appearance is a pair of electrons drawn as a pair of dots. Other specifications can be chosen. The specification follows the pattern *<angle>**<separator>*. *<angle>* is a positive or negative integer denoting the angle counter clockwise to the horizontal where the electrons should be drawn. *<separator>* is either a dot (., single electron), a colon (:, electron pair), a vertical line (|, electron pair), an o (o, empty pair), or a comma (, default spec).

`\chlewis`{*<angle1>**<type1>**<angle2>**<type2>*}{*<atom>*}

For example: `\chlewis{0,180}{0}` gives :O: and `\chlewis{0.90.180.270.}{C}` gives $\text{•}\overset{\cdot}{\underset{\cdot}{\text{C}}}\text{•}$.

The appearance can be influenced by a number of options:

`lewis-default` = . | : || | o | single | pair | pair (dotted) | pair (line) | empty Default: pair
Sets the default type that is used when no type is given in *<electron spec>*.

`lewis-distance` = {*<dim>*} Default: 1ex
The distance of two electrons in a pair.

`lewis-line-length` = {*<dim>*} Default: 1.5ex
The length of the line representing an electron pair.

`lewis-line-width` = {*<dim>*} Default: 1pt
The thickness of a line representing an electron pair.

`lewis-offset` = {*<dim>*} Default: .5ex
The distance of the symbols from the atom.


The dots are drawn according to the `radical-radius` option mentioned in section 7.6.

The basic usage should be more or less self-explaining:

```

1 \chlewis{0:90|180.270}{O}
2 \quad
3 \chlewis{45,135}{O}
4 \quad
5 \chlewis{0o}{Na}

```

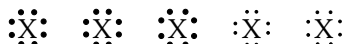


The next example shows the effect of some of the options:

```

1 \chlewis[lewis-default=.]{23,68,113,158,203,248,293,338}{X}
2 \quad
3 \chlewis{0,90,180,270}{X}
4 \quad
5 \chlewis[lewis-distance=1.25ex]{0,90,180,270}{X}
6 \quad
7 \chlewis[lewis-distance=.75ex,radical-radius=.5pt]{0,90,180,270}{X}
8 \quad
9 \chlewis[
10   radical-radius=.5pt,
11   lewis-default=.
12 ]{23,68,113,158,203,248,293,338}{X}

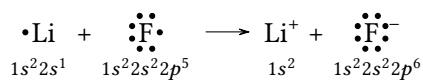
```



```

1 \ch{
2   !($1s^22s^1$)( "\chlewis{180.}{Li}" ) +
3   !($1s^22s^22p^5$)( "\chlewis{0.90,180,270}{F}" )
4   ->
5   !($1s^2$)( Li+ ) + !($1s^22s^22p^6$)( "\chlewis{0,90,180,270}{F}" {}- )
6 }

```



16 Kröger-Vink Notation

Introduced in
version 4.5

CHEMFORMULA also supports the Kröger-Vink notation.

References

`kroeger-vink = true|false`

Default: false

Enable the Kröger-Vink notation. As most options this can be enabled globally via the setup command or locally as option to `\ch`.

With this option enabled several changes come into effect: ' produces a prime, a x in a superscript produces \times , and both a . and a * produce a little filled circle. In the Kröger-Vink notation a prime denotes a negative relative charge, the circle a positive relative charge, and the cross denotes a neutral relative charge.

<pre> 1 \setchemformula{kroeger-vink=true} 2 \ch{Al_{Al}^{'}} 3 \ch{Al_{Al}^{'}}\par 4 \ch{Ni_{Cu}^{\times}}\par 5 \ch{V_{Cl}^{\bullet}} 6 \ch{V_{Cl}^{\bullet*}}\par 7 \ch{Ca_i^{\bullet\bullet}}\par 8 \ch{e^{'}}\par 9 \ch{Cl_i^{'}} 10 \ch{Cl_i^{'}}\par 11 \ch{O_i^{'\prime}} 12 \ch{O_i^{'\prime}}</pre>	$Al'_{Al} Al'_{Al}$ Ni^{\times}_{Cu} $V^{\bullet}_{Cl} V^{\bullet}_{Cl}$ $Ca_i^{\bullet\bullet}$ e' $Cl'_i Cl'_i$ $O_i'' O_i''$
--	---

There are a number of options for customizations:

`kv-positive-style = {<TikZ>}` (initially empty)
TikZ code for positive charge dot.

`kv-positive-radius = {<dim>}` Default: .3ex
Radius of positive charge dot

`kv-positive-hshift = {<dim>}` Default: .15em
Horizontal shift of positive charge dot

`kv-positive-vshift = {<dim>}` Default: .5ex
Vertical shift positive charge dot

`kv-positive-offset = {<dim>}` Default: .4em
The offset of two consecutive positive charge dots

`kv-neutral-symbol = {<TeX code>}` Default: \times
Symbol for neutral particles.

References

[Ame02] AMERICAN MATHEMATICAL SOCIETY. `amsmath`. version 2.0, Feb. 25, 2002.
URL: <http://mirror.ctan.org/macros/latex/required/amsmath/>.

References

- [Coh+08] E. Richard COHAN et al.
“*Quantities, Symbols and Units in Physical Chemistry*”, IUPAC Green Book.
3rd Edition. 2nd Printing. IUPAC & RSC Publishing, Cambridge, 2008.
- [Feu13] Christian FEUERSÄNGER. pgfplots. version 1.9, Oct. 4, 2013.
URL: <http://mirror.ctan.org/graphics/pgf/contrib/pgfplots/>.
- [KN12] Markus KOHM and Frank NEUKAHM. KOMA-Script. version 3.11b, July 29, 2012.
URL: <http://mirror.ctan.org/macros/latex/contrib/koma-script/>.
- [Rei98] Axel REICHERT. nicefrac. version 0.9b, Aug. 4, 1998.
URL: <http://mirror.ctan.org/macros/latex/contrib/units/>.
- [Tan13] Till TANTAU. TikZ/pgf. version 3.0.0, Dec. 13, 2013.
URL: <http://mirror.ctan.org/graphics/pgf/>.
- [Tel13] Christian TELLECHEA. chemfig. version 1.0h, Nov. 28, 2013.
URL: <http://mirror.ctan.org/macros/generic/chemfig/>.
- [The13a] THE L^AT_EX₃ PROJECT TEAM. l3kernel. version SVN 4582, July 28, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/l3kernel/>.
- [The13b] THE L^AT_EX₃ PROJECT TEAM. l3packages. version SVN 4582, July 28, 2013.
URL: <http://mirror.ctan.org/macros/latex/contrib/l3packages/>.

Index

Symbols

-/> (arrow type)	23
-> (arrow type)	23
<- (arrow type)	23
<-> (arrow type)	23
</- (arrow type)	23
<<=> (arrow type)	24
<==> (arrow type)	3, 24
<=> (arrow type)	24
<=>> (arrow type)	24
<> (arrow type)	24
<0> (arrow type)	24
== (arrow type)	24
>=< (arrow type)	3, 24
>=<< (arrow type)	3, 24
>>=< (arrow type)	3, 24

A

adduct-space	15, 18
AMERICAN MATHEMATICAL SOCIETY	4
amsmath (package)	4
arrow-min-length	26
arrow-offset	26
arrow-penalty	26
arrow-ratio	26
arrow-style	2, 26, 28
arrow-yshift	26

B

\bond	13 ff., 18 f.
bond-length	15 f.
bond-offset	16
bond-penalty	16
bond-style	16

C

\ce	5
\ch	3-13, 15-25, 27, 29-34, 36 f.
charge-hshift	12, 15 ff.
charge-style	15
charge-vshift	15
\charrow	29
\chcpd	19
chemfig (package)	27, 35
chemformula	4, 6
\chemformula@bondlength	15
\chemsetup	4, 6, 12, 15-18, 23, 32 f.
\chlewis	2, 35 f.
\chname	31
\chstoich	3, 7
circled	3, 12, 15

circletype	3, 12, 15
COHAN, E. Richard	7, 13, 16
compound-sep	26
CTAN	4, 34

D

decimal-marker	7, 11
\DeclareChemArrow	27
\DeclareChemBond	14
\DeclareChemBondAlias	14
\DeclareChemCompoundProperty	4, 19

F

FEUERSÄNGER, Christian	34
font-family	33
font-series	33
font-shape	33
font-spec	33
fontspec (package)	33
format	32
\fplus	12
frac-math-cmd	7
frac-style	7 f.

H

HENSEL, Martin	2
----------------	---

K

KOHN, Markus	4
KOMA-Script (bundle)	4
kroeger-vink	37
kv-neutral-symbol	37
kv-positive-hshift	37
kv-positive-offset	37
kv-positive-radius	37
kv-positive-style	37
kv-positive-vshift	37

L

l3kernel (bundle)	4
l3keysze (package)	4
l3packages (bundle)	4
label-offset	26
label-style	26
lewis-default	35
lewis-distance	35
lewis-line-length	35
lewis-line-width	35
lewis-offset	35
LPPL	4

INDEX

M	
<code>math-space</code>	23
<code>\mch</code>	11
<code>mhchem</code> (package).....	2, 4, 23
<code>minus-output-symbol</code>	3, 21
<code>minus-penalty</code>	3, 20
<code>minus-space</code>	3, 20
N	
<code>name-format</code>	30
<code>name-width</code>	31
NEUKAHM, Frank.....	4
<code>\NewChemArrow</code>	27 ff.
<code>\NewChemBond</code>	14
<code>\NewChemBondAlias</code>	14
<code>\NewChemCompoundProperty</code>	4, 19 f.
<code>nicefrac</code> (package).....	4, 8
O	
<code>\ox</code>	10, 22, 25
P	
<code>pgfmanual</code> (manual).....	28
<code>pgfplots</code> (package).....	34
<code>plus-output-symbol</code>	3, 20
<code>plus-penalty</code>	3, 20
<code>plus-space</code>	3, 20
Q	
“Quantities, Symbols and Units in Physical Chemistry”, <i>IUPAC Green Book</i>	7, 13, 16
R	
<code>radical-hshift</code>	16
<code>radical-radius</code>	16, 35
<code>radical-space</code>	16
<code>radical-style</code>	16
<code>radical-vshift</code>	16
REICHERT, Axel.....	4
<code>\RemoveChemCompoundProperty</code>	4, 19
<code>\RenewChemArrow</code>	27, 29
<code>\RenewChemBond</code>	14
<code>\RenewChemCompoundProperty</code>	4, 19
S	
<code>scrfile</code> (package).....	4
<code>\setchemformula</code>	4, 6, 11, 34, 37
<code>\ShowChemArrow</code>	28 f.
<code>\ShowChemBond</code>	14
<code>stoich-paren-parse</code>	7
<code>stoich-print</code>	3, 7
<code>stoich-space</code>	7 f.
<code>subscript-style</code>	15
<code>subscript-vshift</code>	15, 17
T	
TANTAU, Till.....	4
TELLECHEA, Christian.....	27, 35
THE L ^A T _E X ₃ PROJECT TEAM.....	4
<code>tikz</code> (package).....	4
<code>tikz-external-disable</code>	34
<code>TikZ/pgf</code> (package).....	4
X	
<code>xfrac</code> (package).....	4, 7 f.
<code>xparse</code> (package).....	4