2023-07-12

whalogen v0.1.0

Spencer Chang
spencer@sycee.xyz

Contents

Quick Start Guide

```
#import "@preview/whalogen:0.1.0": ce

This is the formula for water: #ce("H2O")

This is a chemical reaction:
#ce("CO2 + C -> 2CO")

It can be placed inside an equation: $#ce("CO2 + C -> 2CO")$

It can be placed on its own line:
$
#ce("CO2 + C -> 2CO")
$
```

This is the formula for water: H_2O

This is a chemical reaction: $CO_2 + C \longrightarrow 2CO$

It can be placed inside an equation: $CO_2 + C \longrightarrow 2CO$

It can be placed on its own line:

$$CO_2 + C \longrightarrow 2CO$$

Introduction

This package is designed to facillitate the typesetting of chemical formulas in Typst. It mainly aims to replicate the functionality and operation of LaTeX's `mhchem` package. However, there are naturally some differences in the implementation of some features.

All of the features of this package are designed to function in both text and math mode.

Chemical Equations

$CO_2 + C \longrightarrow 2CO$

```
#ce("CO2 + C -> 2CO")
```

Chemical Formulae

H_2O

```
#ce("H2O")
```

Sb_2O_3

```
#ce("Sb2O3")
```

Charges

H^+

```
#ce("H+")
```

CrO_4^{2-}

```
#ce("CrO4^2-")
```

$[AgCl_2]^-$

```
#ce("[AgCl2]-")
```

Y^{99+}

```
#ce("Y^99+")
```

The en-dash (–) will be used for minus.

Oxidation States

$Fe^{II}Fe_2^{III}O_4$

```
#ce("Fe^II Fe^III_2O4")
```

A caret (^) will imply that subsequent characters should be in the superscript unless interrupted by certain characters (i.e. whitespace and underscore).

Stoichiometric Numbers

$2\mathrm{H_2O}$	`#ce("2H2O")`
$2\mathrm{H_2O}$	`#ce("2 H2O")`
$0.5\mathrm{H_2O}$	`#ce("0.5H2O")`
$\frac{1}{2}\mathrm{H_2O}$	`#ce("1/2H2O")`

The behavior of the fraction line is inherited from Typst's default behavior and generally obeys the rules for grouping numbers and automatically removing parenthesis.

When whitespace is inserted between the stoichiometric number and a compound with a subscript, the whitespace is automatically trimmed. If there is no subscript, the whitespace is not trimmed. This is a characteristic of Typst.

$2\ \mathrm{CO}$	`#ce("2 CO")`

Nuclides, Isotopes

${}^{227}_{90}\mathrm{Th}^{+}$	`#ce("@Th,227,90@^+")`
${}^{0}_{-1}\mathrm{n}^{-}$	`#ce("@n,0,-1@^-")`

To simplify implementation of the parser, this string pattern is introduced to specify nuclides and isotopes. This is different from `mhchem`.

Parenthesis, Brackets, Braces

$\mathrm{(NH_4)_2S}$	`#ce("(NH4)2S")`

The parenthesis, brackets, and braces will automatically size to match the inner content.

$\left[\left\{\mathrm{(X_2)_3}\right\}_2\right]^{3+}$	`#ce("[{(X2)3}2]^3+")`

Reaction Arrows

$A \longrightarrow B$	`#ce("A -> B")`
$A \longleftarrow B$	`#ce("A <- B")`
$A \longleftrightarrow B$	`#ce("A <-> B")`
$A \leftrightarrows B$	`#ce("A <--> B")`

These reaction arrows come from the built-in `sym` module.

It is also possible to include text above the arrow. The backend uses the `xarrow` package to resize the arrows and place respective text.

$A \xrightarrow{H_2O} B$	`#ce("A ->[H2O] B")`
$A \xrightarrow{\text{some text}} B$	`#ce("A ->[some text] B")`

States of Aggregation

$H_{2(aq)}$	`#ce("H2(aq)")`
$NaOH(aq, \infty)$	`#ce("NaOH(aq, #sym.infinity)")`

In Typst, when parenthesis and similar follow a subscript, it is also included in the subscript. This results in seemingly different behavior for very similar input as seen above. The H_2 has an implied underscore whereas the $NaOH$ does not.

To achieve the opposite behavior as shown above, insert whitespace or underscore respectively.

$H_2(aq)$	`#ce("H2 (aq)")`
$NaOH_{(aq, \infty)}$	`#ce("NaOH_(aq, #sym.infinity)")`

Further Examples

ce is a function that takes string input and returns a content block. As such, it can interact with the same rules as other content blocks in math mode.

```
Example to calculate $K_a$:
$
K_a = (#h(1.3em) [#ce("H+")] overbrace(#ce("[A-]"), "conjugate acid"))/
underbrace(#ce("[HA]"), "acid")
$
Using `ce` in limit text:
$
lim_#ce("[H+]->#sym.infinity") K_a = #sym.infinity
$
Aligning multiple equations:
$
#ce("H2SO4 (aq) &<-> H+ (aq) + HSO4^- (aq)")\
#ce("H+ (aq) + HSO4^- (aq) &<-> H2SO4 (aq)")
$
```

Example to calculate K_a:

$$K_a = \frac{\quad [\mathrm{H}^+] \; \overbrace{[\mathrm{A}^-]}^{\text{conjugate acid}}}{\underbrace{[\mathrm{HA}]}_{\text{acid}}}$$

Using ce in limit text:

$$\lim_{[\mathrm{H}^+] \longrightarrow \infty} K_a = \infty$$

Aligning multiple equations:

$$\begin{aligned} \mathrm{H_2SO_4(aq)} &\longleftrightarrow \mathrm{H^+(aq) + HSO_4^-(aq)} \\ \mathrm{H^+(aq) + HSO_4^-(aq)} &\longleftrightarrow \mathrm{H_2SO_4(aq)} \end{aligned}$$

Appendix

Under the Hood

This package works by parsing the provided simple text input and converting it into Typst equation input. In general, inserting whitespace will reset the parser's state machine. If there is still some strange behavior, the underlying Typst output can be inspected by setting the debug flag:

```
"H"_"2"("aq")
```

```
#ce("H2(aq)", debug: true)
```