

AsciiMath

Fork me on GitHub

[about](#) [getting started](#) [syntax](#)

About

AsciiMath is an easy-to-write markup language for mathematics.
Try it out in the interactive renderer:

Input:

```
sum_(i=1)^n i^3=((n(n+1))/2)^2
```

Rendering:

$$\sum_{i=1}^n i^3 = \left(\frac{n(n+1)}{2} \right)^2$$

Getting Started

In order to get started you have two options:

1. Use [MathJax](#) to render your formulas. MathJax is a full fledged open source JavaScript display engine for mathematics and **works in all browsers**.
This is the recommended approach!

Get started by loading the default AsciiMath configuration:

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/mathjax/2.7.4/latest.js?config=AM_CHTML"></script>
```

Visit the MathJax [Getting Started](#) page to find other CDN options , or to learn how to install MathJax locally on your server.

Text in your HTML enclosed in ``` (backticks) will now get rendered as a math formula.
The math delimiters can also be customized. Check out the [MathJax](#) website for more information!

2. Load the AsciiMath javascript file ([get it on GitHub](#)) in either the `head` or the `body` tag of your website like this:

```
<script src="ASCIIMathML.js"></script>
```

This file contains JavaScript to convert AsciiMath notation and (some) LaTeX to Presentation MathML. The conversion is done while the HTML page loads.

Attention: Currently this only works in Firefox and Safari.

While HTML5 now includes MathML as an official recommendation, the remaining browsers do not appear to be implementing it. For widest browser compatibility, the use of MathJax is recommended.

Syntax

Most AsciiMath symbols attempt to mimic in text what they look like rendered, like `oo` for ∞ . Many symbols can also be displayed using a TeX alternative, but a preceeding backslash is not required.

Operation symbols

Type	TeX alt	See
+		+
-		—
*	cdot	·
**	ast	*
***	star	★
//		/
\	backslash setminus	\
xx	times	×
÷	div	÷
><	ltimes	⋈
><	rtimes	⋉
><	bowtie	⋈
@	circ	◦
o+	oplus	⊕
ox	otimes	⊗
o.	odot	⊙
sum		Σ
prod		Π
^^	wedge	∧
^^^	bidwedge	⋀
vv	vee	∨
vvv	bigvee	⋃
nn	cap	∩
nnn	bigcap	⋂
uu	cup	∪
uuu	bigcup	⋃

Miscellaneous symbols

Type	TeX alt	See
2/3	frac{2}{3}	$\frac{2}{3}$
2^3		2³
sqrt x		\sqrt{x}
root(3)(x)		$\sqrt[3]{x}$
int		∫
oint		∮
del	partial	∂
grad	nabla	∇
+ -	pm	±
O/	emptyset	∅
oo	infty	∞
aleph		ℵ
∴	therefore	∴
∵	because	∵
...	ldots	...
cdots		...
vdots		⋮
ddots		⋱
quad		
/_	angle	∠
frown		⤿
/\	triangle	△
diamond		◇
square		□
_	lfloor	⌊
_	rfloor	⌋
~	lceil	⌈
~	rcell	⌉

Relation symbols

Type	TeX alt	See
=		=
!=	ne	≠
<	lt	<
>	gt	>
<=	le	≤
>=	ge	≥
-<	prec	⋈
-<=	preceq	⋈
>-	succ	⋈
>=	succeq	⋈
in		∈
!in	notin	∉
sub	subset	⊂
sup	supset	⊃
sube	subseteq	⊆
supe	supseteq	⊇
=	equiv	≡
~	cong	≅
~	approx	≈
prop	propto	∝

CC		\mathbb{C}
NN		\mathbb{N}
QQ		\mathbb{Q}
RR		\mathbb{R}
ZZ		\mathbb{Z}
"hi"	text(hi)	hi

Logical symbols

Type	TeX alt	See
and		and
or		or
not	neg	\neg
=>	implies	\Rightarrow
if		if
<=>	iff	\Leftrightarrow
AA	forall	\forall
EE	exists	\exists
\bot	bot	\bot
TT	top	\top
--	vdash	\vdash
==	models	\models

Grouping brackets

Type	TeX alt	See
((
))
[[
]]
{		{
}		}
(:	langle	\langle
:)	rangle	\rangle
<<		\langle
>>		\rangle
{: x)		$x)$
(x :}		$(x$
abs(x)		$ x $
floor(x)		$\lfloor x \rfloor$
ceil(x)		$\lceil x \rceil$
norm(vecx)		$\ \vec{x}\ $

Arrows

Type	TeX alt	See
uarr	uparrow	\uparrow
darr	downarrow	\downarrow
rarr	rightarrow	\rightarrow
->	to	\rightarrow
>->	rightarrowtail	\rightarrowtail
->>	twoheadrightarrow	\twoheadrightarrow
>->>	twoheadrightarrowtail	\twoheadrightarrowtail
->	mapsto	\mapsto
larr	leftarrow	\leftarrow
harr	leftrightarrow	\leftrightarrow
rArr	Rightarrow	\Rightarrow
lArr	Leftarrow	\Leftarrow
hArr	Leftrightarrow	\Leftrightarrow

Accents

Type	TeX alt	See
hat x		\hat{x}
bar x	overline x	\bar{x}
ul x	underline x	\underline{x}
vec x		\vec{x}
dot x		\dot{x}
ddot x		\ddot{x}
overset(x)(=)	overset(x)(=)	$\overset{x}{=}$
underset(x)(=)		$\underset{x}{=}$
ubrace(1+2)	underbrace(1+2)	$\underbrace{1+2}$
obrace(1+2)	overbrace(1+2)	$\overbrace{1+2}$
color(red)(x)		$\textcolor{red}{x}$
cancel(x)		\cancel{x}

Greek Letters

Type	See	Type	See
alpha	α		
beta	β		
gamma	γ	Gamma	Γ
delta	δ	Delta	Δ
epsilon	ε		
varepsilon	ϵ		
zeta	ζ		
eta	η		
theta	θ	Theta	Θ
vartheta	ϑ		
iota	ι		
kappa	κ		
lambda	λ	Lambda	Λ
mu	μ		
nu	ν		
xi	ξ	Xi	Ξ
pi	π	Pi	Π
rho	ρ		
sigma	σ	Sigma	Σ
tau	τ		
upsilon	υ		
phi	ϕ	Phi	Φ
varphi	φ		
chi	χ		
psi	ψ	Psi	Ψ
omega	ω	Omega	Ω

Font commands

Type	See
bb "AaBbCc"	AaBbCc
bbb "AaBbCc"	AaBbCc
cc "AaBbCc"	<i>AaBbCc</i>
tt "AaBbCc"	AaBbCc
fr "AaBbCc"	<i>AaBbCc</i>
sf "AaBbCc"	AaBbCc

Standard Functions

sin, cos, tan, sec, csc, cot, arcsin, arccos, arctan, sinh, cosh, tanh, sech, csch, coth, exp, log, ln, det, dim, mod, gcd, lcm, lub, glb, min, max, f, g.

Special Cases

Matrices: `[[a,b],[c,d]]` yields to $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

Column vectors: `((a),(b))` yields to $\begin{pmatrix} a \\ b \end{pmatrix}$

Matrices can be used for layout: `{(2x,+,17y,=,23),(x,-,y,=,5):}` yields
$$\begin{cases} 2x + 17y = 23 \\ x - y = 5 \end{cases}$$

Complex subscripts: `lim_(N->oo) sum_(i=0)^N` yields to $\lim_{N \rightarrow \infty} \sum_{i=0}^N$

Subscripts must come before superscripts: `int_0^1 f(x)dx` yields to $\int_0^1 f(x)dx$

Derivatives: `f'(x) = dy/dx` yields $f'(x) = \frac{dy}{dx}$

For variables other than x,y,z, or t you will need grouping symbols: `(dq)/(dp)` for $\frac{dq}{dp}$

Overbraces and underbraces: `ubrace(1+2+3+4)_("4 terms")` yields $\underbrace{1 + 2 + 3 + 4}_{4 \text{ terms}}$.

`obrace(1+2+3+4)^("4 terms")` yields $\overbrace{1 + 2 + 3 + 4}^{4 \text{ terms}}$.

Attention: Always try to surround the `>` and `<` characters with spaces so that the html parser does not confuse it with an opening or closing tag!

The Grammar

Here is a definition of the grammar used to parse AsciiMath expressions. In the Backus-Naur form given below, the letter on the left of the `::=` represents a category of symbols that could be one of the possible sequences of symbols listed on the right. The vertical bar `|` separates the alternatives.

```
v ::= [A-Za-z] | greek letters | numbers | other constant symbols
u ::= sqrt | text | bb | other unary symbols for font commands
b ::= frac | root | stackrel | other binary symbols
l ::= ( | [ | { | (: | {: | other left brackets
r ::= ) | ] | } | :) | :} | other right brackets
S ::= v | lEr | uS | bSS           Simple expression
I ::= S_S | S^S | S_S^S | S       Intermediate expression
E ::= IE | I/I                     Expression
```