HyperSnips Snippet

June 18, 2025

Snippet

		Context
table([1-9]1) ([1-9]1)	Table environment	notmath(context)
ary([1-9]1) ([1-9]1)	Array environment	math(context)
(b|p)mat([1-9]1) ([1-9]1)	matrix	math(context)
wrt	with respect to	notmath(context)
iid	identical and independently distributed	notmath(context)
(?<![a-zA-Z])opn	operator	math(context)
wp	with probability	none
(?<![a-zA-Z])(;m|mu)	μ	math(context)
(?<![a-zA-Z])(;a|alpha)	α	math(context)
(?<![a-zA-Z])(;s|sigma)	σ	math(context)
(?<![a-zA-Z])(;S|Sigma)	Σ	math(context)
(?<![a-zA-Z])(;r|rho)	ρ	math(context)
(?<![a-zA-Z])(;b|beta)	β	math(context)
(?<![a-zA-Z])(;g|gamma)	γ	math(context)
(?<![a-zA-Z])(;G|Gamma)	Γ	math(context)
(?<![a-zA-Z])(;d|delta)	δ	math(context)
(?<![a-zA-Z])(;D|Delta)	Δ	math(context)
(?<![a-zA-Z])(;z|zeta)	ζ	math(context)
(?<![a-zA-Z])(;e|eta)	η	math(context)
(?<![a-zA-Z])(;t|theta)	θ	math(context)
(?<![a-zA-Z])(;vt|vartheta)	ϑ	math(context)
(?<![		

		Context
a-zA-Z])(;T|Theta)	Θ	math(context)
(?<![		
a-zA-Z])(;;t|tau)	τ	math(context)
(?<![		
a-zA-Z])(;i|iota)	ι	math(context)
(?<![		
a-zA-Z])(;k|kappa)	κ	math(context)
(?<![		
a-zA-Z])(;l|lambda)	λ	math(context)
(?<![		
a-zA-Z])(;L|Lambda)	Λ	math(context)
(?<![		
a-zA-Z])(;n|nu)	ν	math(context)
(?<![		
a-zA-Z])(;;n|nable)	∇	math(context)
(?<![		
a-zA-Z])(;p|pi)	π	math(context)
(?<![		
a-zA-Z])(;P|Pi)	Π	math(context)
(?<![		
a-zA-Z])(;r|rho)	ρ	math(context)
(?<![		
a-zA-Z])(;t|tau)	τ	math(context)
(?<![		
a-zA-Z])(;u|upsilon)	υ	math(context)
(?<![		
a-zA-Z])(;U|Upsilon)	Υ	math(context)
(?<![		
a-zA-Z])(;;p|phi)	ϕ	math(context)
(?<![		
a-zA-Z])(;;P|Phi)	Φ	math(context)
(?<![		
a-zA-Z])(;vp|varphi)	φ	math(context)
(?<![		
a-zA-Z])(;c|chi)	χ	math(context)
(?<![		
a-zA-Z])(;;;p|psi)	ψ	math(context)
(?<![		
a-zA-Z])(;;;P|Psi)	Ψ	math(context)
(?<![		
a-zA-Z])(;o|omega)	ω	math(context)
(?<![		
a-zA-Z])(;O|Omega)	Ω	math(context)
(?<![		
a-zA-Z])(;x|xi)	ξ	math(context)
(?<![		
a-zA-Z])(;X|Xi)	Ξ	math(context)
eps	ϵ	math(context)
veps		math(context)
(?<![		
a-zA-Z])ell	ℓ	math(context)
//	Fraction	math(context)

		Context
((+)|(*)( )?([A-Za-z]+)((|_)({+}|))*)/	Fraction no ()	math(context)
`*/	Fraction with ()	none
>=|(?<!)geq	greater or equal to	math(context)
<	less or equal to (up to constant)	math(context)
>	less or equal to (up to constant)	math(context)
<=|(?<!)leq	less or equal to	math(context)
!=|(?<!)neq	not equal	math(context)
|=|(?<!)mdl	models	math(context)
|−|(?<!)vdh	models	math(context)
suc	succeeds	math(context)
seq	succeeds or equal to	math(context)
prec	precedes	math(context)
peq	precedes or equal to	math(context)
==	constan equal	math(context)
	approx equal	math(context)
=	approx equal(2)	math(context)
-	approx equal(3)	math(context)
(?<!)cir	composition	math(context)
@>	inclusion	math(context)
||	mid	math(context)
([a-zA-Z])	subscript	math(context)
([A-Za-z]])($[A-Za-z+][A-Za-z-+])$	auto subscript	math(context)
([A-Za-z]])`([A-Za-z+][A-Za-z-+])	auto supscript	math(context)
(b|B)(ar)	bar	math(context)
(?[a-zA-Z]*|(>(+})(b|B)(ar)	bar	math(context)
(t|T)(d)	tilde	math(context)
(?[a-zA-Z]*|(>(+})(t|T)(d)	tilde	math(context)
(?[a-zA-Z]*|(>(+})(h|H)(t)	hat	math(context)
(?[a-zA-Z]*|(>(+})(b|B)(f)	mathbf	math(context)
(?[a-zA-Z]*|(>(+})(b|B)(m)	bm	math(context)
(		

		Context
?[a-zA-Z]*|(>(+})(,¦,)	Vector postfix	math(context)
1..n	sequence	math(context)
fig	Figure environment	notmath(context)
...	dots	math(context)
=>	implies	math(context)
=<	implied by	math(context)
(?<!)iff	if and only if	math(context)
fm	inline Math	none
dm	display Math	none
<>	<>	math(context)
lr,	<>	math(context)
lrd	()	math(context)
		math(context)
lra		math(context)
lrq	[]	math(context)
ceil	ceil	math(context)
Ceil	ceil	math(context)
flr	floor	math(context)
Flr	floor	math(context)
abs	abs value	math(context)
Abs	abs value	math(context)
norm	norm	math(context)
Norm	Norm	math(context)
(?<!)sum	sum	math(context)
Sum	sum	math(context)
(?<!)scup	disjoint union	math(context)
(?<!)cup	union	math(context)
Cup	Union	math(context)
(?<!)cap	intersection	math(context)
Cap	Intersection	math(context)
Conj	conjunction	math(context)
Disj	disjunction	math(context)
(?<!)sub	subset	math(context)
(?<!)nsub	nsubseteq	math(context)
sube	subseteq	math(context)
subn	subsetneq	math(context)
sups	supset	math(context)
(?<!)nsup	nsupseteq	math(context)
supe	supseteq	math(context)

		Context
supn	supsetneq	math(context)
(?<!)nlim	nolimit	math(context)
(?<!)lim	limit	math(context)
(?<!)lsup	limsup	math(context)
(?<!)linf	liminf	math(context)
(?<!)prd	product	math(context)
Prd	product	math(context)
(?<!)coprd	coproduct	math(context)
pt	∂	math(context)
pdif	d/dx	math(context)
dif	d/dx	math(context)
(?<!)sq	$\sqrt{}$	math(context)
oo	∞	math(context)
ôo	supscript ∞	math(context)
EE	exist	math(context)
AA	forall	math(context)
([a-zA-Z])([a-zA-Z])()	?[?+?]	math(context)
o	oplus	math(context)
o(x|ẋ)	otimes	math(context)
(x|ẋ)	times	math(context)
split	split	math(context)
case	cases	math(context)
tet	text	math(context)
( )([a-zA-Z])' A "rv = m[2].toUpperCase()" 0*endsnippet* context math(context) snippet nin iA $\notin$0 endsnippet context math(context) snippet inv	inverse	math(context)
(?<!)tp	transpose	math(context)
prp	perp	math(context)
cp	complement	math(context)
qs	square	math(context)
->|(?<!)to	to	math(context)
<->	to	math(context)
!>	mapsto	math(context)
dint	integral	math(context)
(?<!		

		Context
)int	integral without anything	math(context)
(?<!)not	lnot	math(context)
(?<!)(sin|cos|tan|cot|csc|sec|ln|log|exp|gcd|perp|inf|sup|Tr|diag|rank|det|dim|ker|Im|Re|dom|arg|min|max|mod|cot|land|lor)		math(context)
(?<!)a(sin|cos|tan|cot|csc|sec)	arc-trigonometry	math(context)
a(max|min)	arg min/max	math(context)
'	prime	math(context)
prime} ´	prime	math(context)
–	setminus	math(context)
([a-zA-Z])(c|C)al	mathcal	math(context)
(?+)(b|B)f	textbf	notmath(context)
rm	mathrm	math(context)
([a-zA-Z])(s|S)cr	mathscr	math(context)
(?<!)st	star	math(context)
**	asterisk	math(context)
_*	asterisk	math(context)
ast}	asterisk	math(context)
.	.	math(context)
dot.	..	math(context)
»	»	math(context)
«	«	math(context)
ind	indicator function	math(context)
spt	support	math(context)
mean	Expectation	math(context)
(?<!)Var	Variation	math(context)
(?<!)Cov	Covariance	math(context)
(?<!)Pr	Probability	math(context)
(?<!)sim	sim	math(context)
apx	approx	math(context)
(?<!)bino	binomial coefficient	math(context)
ems	empty set	math(context)
(?<!)emph	emph	none
begg	begin custom env	none
:=	definition in math	math(context)
=:	definition in math	math(context)
::	colon	math(context)
idd	identity	math(context)
(?<!)quo		math(context)
‖		

6

		Context
		math(context)
atf		notmath(context)
hpr	[]	none
lbl		none
rmk	remark	notmath(context)
dfn	definition	notmath(context)
vph	same level of underbrace	math(context)
hom	homorphism	math(context)
Obj	object	math(context)
mor	morphism	math(context)
__		math(context)
`		math(context)
fk		math(context)
tg	△	math(context)
(?<!)qed		notmath(context)
opmin	Optimization problem	notmath(context)
opmax	Optimization problem	notmath(context)
opPD notmath(context)	Optimization problem with primal	dual

Table environment (table([1-9]1) ([1-9]1))

Context: notmath(context)
Flags: bA

```
\begin{table}[H]
        \centering
        \begin{tabular}{``
        let len = m[2];
        let results = "";
        for(var i=0; i<len-1; i++){
                results += "c|"
        }
        results += "c";
        rv = results;
        ``}
            \toprule
                        ``
                    let order = 1;
                    nrow = m[1];
                    ncol = m[2];
                    for (var i=0; i<nrow-1; i++){
                            for(var j = 0;j <ncol-1;j++){
                                    rv += `${snip.tabstop(order+3)} & `;
                                    order ++;
                            }
                            rv += ` ${snip.tabstop(order+3)} \\\\`+ `\\`+ `\n`;
                            order ++;
                            if(i == 0){
                                    rv += `            \\midrule` + `\n`;
```

```
                              }
                              rv += `                                          `;
                      }
                      for(var j = 0;j <ncol−1;j++){
                              rv += `${snip.tabstop(order+3)} & `;
                              order ++;
                      }
                      rv += ` ${snip.tabstop(order+3)} \\\\\` + `\\`;
                      ``
              \bottomrule
              \end{tabular}
              \caption{${2:caption}}
              \label{tab:${3:label}}
      \end{table}
```

Array environment (ary([1-9]1) ([1-9]1))

Context: math(context)
Flags: ibA

```
\begin{array}{``
          let len = m[2];
          let results = "";
          for(var i=0; i<len; i++){
                  results += "c"
          }
          rv = results;
          ``}
          ``
          let order = 1;
          let nrow = m[1];
          let ncol = m[2];
          for (var i=0; i<nrow−1; i++){
                  for(var j = 0;j <ncol−1;j++){
                          rv += `${snip.tabstop(order)} & `;
                          order ++;
                  }
                  rv += ` ${snip.tabstop(order)} \\\\\`+ `\\`+ `\n` + `         `;
                  order ++;
          }
          for(var j = 0;j <ncol−1;j++){
                  rv += `${snip.tabstop(order)} & `;
                  order ++;
          }
          rv += ` ${snip.tabstop(order)} \\\\\`+ `\\`;
          ``
\end{array}
```

matrix ((b|p)mat([1-9]1) ([1-9]1))

Context: math(context)
Flags: iwA

```
\begin{''rv = m[1]''matrix}''
        let order = 1;
        let nrow = m[2];
        let ncol = m[3];
        rv = '\n';
        for (var i=0; i<nrow; i++){
                rv += ' ';
                for(var j = 0;j <ncol−1;j++){
                        rv += '${snip.tabstop(order)} & ';
                        order ++;
                }
                rv += ' ${snip.tabstop(order)} \\\\'+ '\\'+ '\n';
                order ++;
        }
        ''\end{''rv = m[1]''matrix}$0
```

with respect to (wrt)

Context: notmath(context)
Flags: wA

```
w.r.t.\ $0
```

identical and independently distributed (iid)

Context: notmath(context)
Flags: wA

```
i.i.d.\ $0
```

operator ((?<![a-zA-Z])opn)

Context: math(context)
Flags: A

```
\operatorname{$1}($2) $0
```

with probability (wp)

Context: none
Flags: wA

```
w.p.\ $0
```

$\mu((? <![a-zA-Z])(;m|mu))$

Context: math(context)
Flags: A

\mu $0

$\alpha((? <![$
$a - zA - Z])(; a|alpha))$

Context: math(context)
Flags: A

\alpha $0

$\sigma((? <![$
$a - zA - Z])(; s|sigma))$

Context: math(context)
Flags: A

\sigma $0

$\Sigma((? <![$
$a - zA - Z])(; S|Sigma))$

Context: math(context)
Flags: A

\Sigma $0

$\rho((? <![$
$a - zA - Z])(; r|rho))$

Context: math(context)
Flags: A

\rho $0

$\beta((? <![$
$a - zA - Z])(; b|beta))$

Context: math(context)
Flags: A

\beta $0

$\gamma((? <![$
$a - zA - Z])(; g|gamma))$

Context: math(context)
Flags: A

\gamma $0

$\Gamma((? <![$
$a - zA - Z])(; G|Gamma))$

Context: math(context)
Flags: A

\Gamma $0

$\delta((? <![$
$a - zA - Z])(; d|delta))$

Context: math(context)
Flags: A

\delta $0

$\Delta((? <![$
$a - zA - Z])(; D|Delta))$

Context: math(context)
Flags: A

\Delta $0

$\zeta((? <![$
$a - zA - Z])(; z|zeta))$

Context: math(context)
Flags: A

\zeta $0

$\eta((? <![$
$a - zA - Z])(; e|eta))$

Context: math(context)
Flags: A

\eta $0

$\theta((? <![$
$a - zA - Z])(; t|theta))$

Context: math(context)
Flags: A

\theta $0

$\vartheta((? <![$
$a - zA - Z])(; vt|vartheta))$

Context: math(context)
Flags: A

$\backslash$**vartheta** $0

$\Theta((? <![$
$a - zA - Z])(; T|Theta))$

Context: math(context)
Flags: A

$\backslash$**Theta** $0

$\tau((? <![$
$a - zA - Z])(; ; t|tau))$

Context: math(context)
Flags: A

$\backslash$**tau** $0

$\iota((? <![$
$a - zA - Z])(; i|iota))$

Context: math(context)
Flags: A

$\backslash$**iota** $0

$\kappa((? <![$
$a - zA - Z])(; k|kappa))$

Context: math(context)
Flags: A

$\backslash$**kappa** $0

$\lambda((? <![$
$a - zA - Z])(; l|lambda))$

Context: math(context)
Flags: A

$\backslash$**lambda** $0

$\Lambda((? <![$
$a-zA-Z])(;L|Lambda))$

Context: math(context)
Flags: A

\Lambda $0

$\nu((? <![$
$a-zA-Z])(;n|nu))$

Context: math(context)
Flags: A

\nu $0

$\nabla((? <![$
$a-zA-Z])(;;n|nable))$

Context: math(context)
Flags: iA

\nabla $0

$\pi((? <![$
$a-zA-Z])(;p|pi))$

Context: math(context)
Flags: A

\pi $0

$\Pi((? <![$
$a-zA-Z])(;P|Pi))$

Context: math(context)
Flags: A

\Pi $0

$\rho((? <![$
$a-zA-Z])(;r|rho))$

Context: math(context)
Flags: A

\rho $0

$\tau((? <![$
$a - zA - Z])(; t|tau))$

Context: math(context)
Flags: A

\tau $0

$\upsilon((? <![$
$a - zA - Z])(; u|upsilon))$

Context: math(context)
Flags: A

\upsilon $0

$\Upsilon((? <![$
$a - zA - Z])(; U|Upsilon))$

Context: math(context)
Flags: A

\Upsilon $0

$\phi((? <![$
$a - zA - Z])(; ; p|phi))$

Context: math(context)
Flags: A

\phi $0

$\Phi((? <![$
$a - zA - Z])(; ; P|Phi))$

Context: math(context)
Flags: A

\Phi $0

$\varphi((? <![$
$a - zA - Z])(; vp|varphi))$

Context: math(context)
Flags: A

\varphi $0

$\chi((? <![$
$a-zA-Z])(;c|chi))$

Context: math(context)
Flags: A

χ $0

$\psi((? <![$
$a-zA-Z])(;;;p|psi))$

Context: math(context)
Flags: A

ψ $0

$\Psi((? <![$
$a-zA-Z])(;;;P|Psi))$

Context: math(context)
Flags: A

Ψ $0

$\omega((? <![$
$a-zA-Z])(;o|omega))$

Context: math(context)
Flags: A

ω $0

$\Omega((? <![$
$a-zA-Z])(;O|Omega))$

Context: math(context)
Flags: A

Ω $0

$\xi((? <![$
$a-zA-Z])(;x|xi))$

Context: math(context)
Flags: A

ξ $0

$\Xi((? <![$
$a-zA-Z])(;X|Xi))$

Context: math(context)
Flags: A

\\ **Xi** $0

$\epsilon(eps)$

Context: math(context)
Flags: iA

\\ **epsilon** $0

(veps)

Context: math(context)
Flags: iA

\\ **varepsilon** $0

$\ell((? <![$
$a-zA-Z])ell)$

Context: math(context)
Flags: A

\\ **ell** $0

Fraction (//)

Context: math(context)
Flags: iA

\\\frac{$1}{$2}$0

Fraction no () (((+)|(*)(
)?([A-Za-z]+)((_)({+}|))*)/)

Context: math(context)
Flags: A

\frac{``rv = m[1]``}{$1}$0

Fraction with () (⋅∗/)

Context: none
Flags: Am

``
```
let str = m[0];
str = str.slice(0, -1);
let lastIndex = str.length - 1;

let depth = 0;
let i = str.length - 1;

while (true) {
    if (str[i] == ')') depth += 1;
    if (str[i] == '(') depth -= 1;
    if (depth == 0) break;
    i -= 1;
}

let results = str.slice(0, i) + "\\frac{" + str.slice(i+1, -1) + "}";
rv = results;
    rv += `{${snip.tabstop(1)}}${snip.tabstop(0)}`;
``
```

greater or equal to (>=|(?<!)geq)

Context: math(context)
Flags: A

\geq $0

less or equal to (up to constant) (<)

Context: math(context)
Flags: A

\lesssim $0

less or equal to (up to constant) (>)

Context: math(context)
Flags: A

\gtrsim $0

less or equal to ($<=|(?<!$ $)$leq)

Context: math(context)
Flags: A

\leq $0

not equal ($!=|(?<!$ $)$neq)

Context: math(context)
Flags: A

\neq $0

models ($\| = |(? <!$ $)mdl$)

Context: math(context)
Flags: A

\models $0

models ($\| - |(? <!$ $)vdh$)

Context: math(context)
Flags: A

\vdash $0

succeeds (suc)

Context: math(context)
Flags: iA

\succ $0

succeeds or equal to (seq)

Context: math(context)
Flags: iA

\succeq $0

precedes (prec)

Context: math(context)
Flags: iA

prec $0

precedes or equal to (peq)

Context: math(context)
Flags: iA

preceq $0

constan equal (==)

Context: math(context)
Flags: iA

equiv $0

approx equal ( )

Context: math(context)
Flags: iA

\thickapprox $0

approx equal(2) (=)

Context: math(context)
Flags: iA

cong $0

approx equal(3) (-)

Context: math(context)
Flags: iA

simeq $0

composition ((?<!
)cir)

Context: math(context)
Flags: A

circ $0

inclusion (@>)

Context: math(context)
Flags: iA

\hookrightarrow $0

mid (||)

Context: math(context)
Flags: A

\mid $0

subscript (([a-zA-Z]))

Context: math(context)
Flags: A

_{``rv = m[1]``} $0

auto subscript (([A-Za-z)

])
$([A - Za - z+][A - Za - z - +]))$**Context** $: math(context)$
Flags $: A$

auto supscript (([A-Za-z)

])([A-Za-z+][A-Za-z-+])) **Context**: math(context)
Flags: A

`` rv = m[1]``^{``rv = m[2]``$1} $0

bar ((b|B)(ar))

Context: math(context)
Flags: A

\overline{$1} $0

bar ((
?[a-zA-Z]*|
(
+})(b|B)(ar))

Context: math(context)
Flags: A

\overline{``rv = m[1]``} $0

tilde ((t|T)(d))

Context: math(context)
Flags: A

\widetilde{$1} $0

tilde ((
?[a-zA-Z]*|
(
+})(t|T)(d))

Context: math(context)
Flags: A

\widetilde{``rv = m[1]``} $0

hat ((
?[a-zA-Z]*|
(
+})(h|H)(t))

Context: math(context)
Flags: A

\hat{``rv = m[1]``} $0

mathbf ((
?[a-zA-Z]*|
(
+})(b|B)(f))

Context: math(context)
Flags: A

\mathbf{``rv = m[1]``} $0

bm ((
?[a-zA-Z]*|
(
+})(b|B)(m))

Context: math(context)
Flags: A

\bm{``rv = m[1]``} $0

Vector postfix ((?[a-zA-Z]*|(+})(;|;))

Context: math(context)
Flags: A

```
\vec{``rv = m[1]``} $0
```

sequence (1..n)

Context: math(context)
Flags: iA

```
$1_1, \dots, $1_n $0
```

Figure environment (fig)

Context: notmath(context)
Flags: b

```
\begin{figure}[${1:H}]
        \centering
        \includegraphics[width=0.8\textwidth]{$2}
        \caption{$3}
        \label{fig:$4}
\end{figure}
```

dots (...)

Context: math(context)
Flags: iA

```
\dots $0
```

implies (=>)

Context: math(context)
Flags: iA

```
\implies $0
```

implied by (=<)

Context: math(context)
Flags: iA

```
\impliedby $0
```

if and only if ((?<!
)iff)

Context: math(context)
Flags: A

```
\iff $0
```

inline Math (fm)

Context: none
Flags: wA

```
\(${1}\)``
let test = t[1][0];
if (test != ',' &&  test != '.' &&  test != '-' && test !='?' && test !=' '){
        rv = ' ';
}
else{
        rv = '';
}``$2
```

display Math (dm)

Context: none
Flags: bA

```
\[
        ${1}
\]$0
```

<> (<>)

Context: math(context)
Flags: iA

```
\langle $1 \rangle $0
```

<> (lr,)

Context: math(context)
Flags: iA

```
\left\langle $1 \right\\rangle $0
```

() (lrd)

Context: math(context)
Flags: iA

```
\left( $1 \right) $0
```

()

Context: math(context)
Flags: iA

`\\{ $1 \\} $0`

(lra)

Context: math(context)
Flags: iA

`\\left\\{ $1 \\right\\} $0`

[] (lrq)

Context: math(context)
Flags: iA

`\left[ $1 \right] $0`

ceil (ceil)

Context: math(context)
Flags: iA

`\lceil $1 \rceil $0`

ceil (Ceil)

Context: math(context)
Flags: iA

`\left\lceil $1 \right\rceil $0`

floor (flr)

Context: math(context)
Flags: iA

`\lfloor $1 \rfloor$0`

floor (Flr)

Context: math(context)
Flags: iA

`\left\lfloor $1 \right\rfloor$0`

abs value (abs)

Context: math(context)
Flags: iA

`\vert ${1} \vert $0`

abs value (Abs)

Context: math(context)
Flags: iA

`\left\vert ${1} \right\vert $0`

norm (norm)

Context: math(context)
Flags: iA

`\lVert $1 \rVert $0`

Norm (Norm)

Context: math(context)
Flags: iA

`\left\lVert $1 \right\rVert $0`

sum ((?<!
)sum)

Context: math(context)
Flags: A

`\sum_{${1:i}} $0`

sum (Sum)

Context: math(context)
Flags: wA

`\sum_{${1:i}=${2:1}}^{${3:\infty}} $0`

disjoint union ((?<!
)scup)

Context: math(context)
Flags: wA

`\sqcup $0`

union ((?<!
)cup)

Context: math(context)
Flags: wA

\cup $0

Union (Cup)

Context: math(context)
Flags: wA

\bigcup_{${1:i}=${2:1}}^{${3:\infty}} $0

intersection ((?<!
)cap)

Context: math(context)
Flags: A

\cap $0

Intersection (Cap)

Context: math(context)
Flags: wA

\bigcap_{${1:i}=${2:1}}^{${3:\infty}} $0

conjunction (Conj)

Context: math(context)
Flags: wA

\bigwedge_{${1:i}=${2:1}}^{${3:\infty}} $0

disjunction (Disj)

Context: math(context)
Flags: wA

\bigvee_{${1:i}=${2:1}}^{${3:\infty}} $0

subset ((?<!)sub)

Context: math(context)
Flags: A

\subset $0

nsubseteq ((?<!)nsub)

Context: math(context)
Flags: A

\nsubseteq $0

subseteq (sube)

Context: math(context)
Flags: iA

\subseteq $0

subsetneq (subn)

Context: math(context)
Flags: iA

\subsetneq $0

supset (sups)

Context: math(context)
Flags: A

\supset $0

nsupseteq ((?<!)nsup)

Context: math(context)
Flags: A

\nsupseteq $0

supseteq (
supe)

Context: math(context)
Flags: A

\ **supseteq** $0

supsetneq (
supn)

Context: math(context)
Flags: A

\ supsetneq $0

nolimit ((?<!
)nlim)

Context: math(context)
Flags: A

\ **nolimits**

limit ((?<!
)lim)

Context: math(context)
Flags: A

\ **lim**_{${1:n} \ **to** ${2:\ **infty**}}

limsup ((?<!
)lsup)

Context: math(context)
Flags: iA

\ **limsup**_{${1:n} \ **to** ${2:\ **infty**}}

liminf ((?<!
)linf)

Context: math(context)
Flags: iA

\ **liminf**_{${1:n} \ **to** ${2:\ **infty**}}

product ((?<!
)prd)

Context: math(context)
Flags: iA

`\prod $0`

product (Prd)

Context: math(context)
Flags: iA

`\prod_{${1:n}=${2:1}}^{${3:\infty}} $0`

coproduct ((?<!
)coprd)

Context: math(context)
Flags: iA

`\coprod_{${1:n}=${2:1}}^{${3:\infty}} $0`

∂ (pt)

Context: math(context)
Flags: iA

`\partial $0`

d/dx (pdif)

Context: math(context)
Flags: iA

`\frac{\partial ${1:V}}{\partial ${2:x}} $0`

d/dx (dif)

Context: math(context)
Flags: iA

`\frac{\mathrm{d}${1:y}}{\mathrm{d}${2:x}} $0`

$\sqrt{}((?<!$
$)sq)$

Context: math(context)
Flags: A

$\backslash\mathbf{sqrt}\{\$\{1\}\}$ $\$0$

$\infty(oo)$

Context: math(context)
Flags: iA

$\backslash\mathbf{infty}$

supscript $^\infty(o)$

Context: math(context)
Flags: A

$\hat{}\{\backslash\mathbf{infty}\}$ $\$0$

exist (EE)

Context: math(context)
Flags: iA

$\backslash\mathbf{exists}$

forall (AA)

Context: math(context)
Flags: iA

$\backslash\mathbf{forall}$

?[?+?] (([a-zA-Z])([a-zA-Z])())

Context: math(context)
Flags: iA

``rv = m[1]``_{``rv = m[2]``+``rv = m[3]``$1}$0

oplus (o)

Context: math(context)
Flags: A

$\backslash\mathbf{oplus}$ $\$0$

otimes ($o(x|\dot{x})$)

Context: math(context)
Flags: A

```
\otimes $0
```

times (($x|\dot{x}$))

Context: math(context)
Flags: A

```
\times $0
```

split (split)

Context: math(context)
Flags: iA

```
\begin{split}
        $1
\end{split}$0
```

cases (case)

Context: math(context)
Flags: iA

```
\begin{dcases}
        $1, &\text{ if } $3 ;\\\\
        $2, &\text{ if } $4 ;\\\\
        $5, &\text{ otherwise} .
\end{dcases}$0
```

text (tet)

Context: math(context)
Flags: iA

```
\text{$1} $0
```

context math(context) snippet nin iA $\notin 0$ endsnippet
context math(context) snippet inv) **Context**: math(context)
Flags: iA

```
^{-1} $0
```

transpose ((?<!)tp)

Context: math(context)
Flags: A

^{\top} $0

perp (prp)

Context: math(context)
Flags: iA

^{\perp} $0

complement (cp)

Context: math(context)
Flags: iA

^{c} $0

square (qs)

Context: math(context)
Flags: iA

^{2} $0

to (->|(?<!)to)

Context: math(context)
Flags: A

\to $0

to (<->)

Context: math(context)
Flags: A

\leftrightarrow $0

mapsto (!>)

Context: math(context)
Flags: iA

\mapsto $0