

# How I'm able to take notes in mathematics lectures using LaTeX and Vim

A while back I answered a question on Quora: Can people actually keep up with note-taking in Mathematics lectures with LaTeX. There, I explained my workflow of taking lecture notes in LaTeX using Vim and how I draw figures in Inkscape. However, a lot has changed since then and I'd like to write a few blog posts explaining my workflow.

I started using LaTeX to write lecture notes in the second semester of my bachelor in mathematics, and I've been using it ever since, which makes for a total of more than 1700 pages of notes. To give you an idea of what those notes look like, here are some examples:

Les 5

$C = \partial D$ . Then by Cauchy's integral formula

$$\begin{aligned} f(z) &= \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{\zeta - z} d\zeta \\ &= \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{(\zeta - z_0 + z_0 - z)} d\zeta \\ &= \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{(\zeta - z_0)} \frac{1}{1 + \frac{z - z_0}{\zeta - z_0}} d\zeta \\ &= \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{(\zeta - z_0)} \frac{1}{1 - \frac{z - z_0}{\zeta - z_0}} d\zeta. \end{aligned}$$

Now,  $z_0$  is center of the disk and  $\zeta$  is on the boundary of the disk.  $z$  is inside the disk. Therefore  $|z - z_0| < |\zeta - z_0|$  and

$$\left| \frac{z - z_0}{\zeta - z_0} \right| \leq r < 1.$$

(Check what variables are moving! TODO) Now as  $r < 1$ , we can write it as a geometric progression:

$$\begin{aligned} &= \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{\zeta - z_0} \sum_{n=0}^{\infty} \left( \frac{z - z_0}{\zeta - z_0} \right)^n d\zeta \\ &= \sum_{n=0}^{\infty} (z - z_0)^n \left( \frac{1}{2\pi i} \int_C \frac{f(\zeta)}{\zeta - z_0} \left( \frac{1}{\zeta - z_0} \right)^n d\zeta \right) \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(z_0)}{n!} (z - z_0)^n. \end{aligned}$$

As a power series converges uniformly on their disk of convergence.  $\square$

**Theorem 16.** Let  $\Omega \subset \mathbb{C}$  be a region,  $f(z)$  a holomorphic function. Let  $\{w_k\}$  a sequence of points in  $\Omega$  such that

- $\{w_k\}$  are distinct
- $w_k \rightarrow z_0 \in \Omega$
- $f(w_k) = 0$  for all  $k$

Then  $f(z) = 0$  for all  $z \in \Omega$ .

Stronger conditions: suppose  $f(z) = 0$  for all  $z \in \ell$ ,  $\ell$  a line inside  $\Omega$ .

**Remark.** In real analysis, this is not the case!

*Proof.* By the previous theorem,  $f(z)$  is analytic at  $z_0$ , i.e.  $\exists r > 0$ ,  $z \in D_r(z_0)$ :

$$f(z) = \sum_{n=0}^{\infty} a_n (z - z_0)^n.$$

CHAPTER 4. CAUCHY THEOREM

27

Les 6

*Proof.* Denote  $F = f - g$ . Let  $D = D_r(z_0)$ . Denote by  $w_k = z_0 + \frac{z_k}{k}$ . Clearly,  $w_k \rightarrow z_0$ ,  $w_k$  are distinct,  $F(w_k) = 0$ . Therefore,  $F \equiv 0$  on  $\Omega$ , therefore  $f(z) = g(z)$  for all  $z \in \Omega$ .  $\square$

Les 5

Let us prove that  $f(z) = 0$  for all  $z$  in some neighbourhood of  $z_0$ . Suppose not. Therefore, there exists a first  $N$  such that  $a_N$  is not 0. Then

$$\begin{aligned} f(z) &= (z - z_0)^N \sum_{n=N}^{\infty} a_n (z - z_0)^{n-N} \\ &= (z - z_0)^N \left( a_N + \sum_{n=N+1}^{\infty} a_n (z - z_0)^{n-N} \right) \\ g(z) &:= \sum_{n=N+1}^{\infty} a_n (z - z_0)^{n-N} \\ f(z) &= (z - z_0)^N (a_N + g(z)) \\ g(z) &\xrightarrow{z \rightarrow z_0} 0 \text{ because of the definition of } g \end{aligned}$$

In particular,  $g(w_k) \xrightarrow{k \rightarrow \infty} 0$

$$\text{And } f(z) = (z - z_0)^N (a_N + g(z)).$$

So  $\exists M : n \geq M \Rightarrow a_N + g(w_k) \neq 0$ .

Then

$$0 = f(w_k) = \underbrace{(w_k - z_0)^N}_{\neq 0} \underbrace{(a_N + g(w_k))}_{\neq 0}.$$

$w_k \neq z_0$  if we look far enough.

This is a contradiction to the fact that  $\exists a_N \neq 0 \Rightarrow f(z) = 0 \quad \forall z$  in some neighbourhood of  $z_0$ .

So we've proved the theorem for a neighbourhood of  $z_0$ . Denote by  $U$  the interior  $\{z \in \Omega \mid f(z) = 0\}$ .  $U \neq \emptyset$ , since  $f(z)$  is zero in some neighbourhood of  $z_0$ .

- $U$  is open, since it is the interior of a set.

- $U$  is closed, let  $z_0$  be a limit point of  $U$ . Therefore,  $\exists$  a sequence of points  $z_k \in U$  and  $z_k \rightarrow z_0$ . We know that  $f(z_k) = 0$ , because  $z_k \in U$ . By the same arguments of above,  $f(z) = 0$ .

Therefore  $V = \Omega \setminus U$  is open and  $\Omega = U \cup V$ . As  $\Omega$  is connected, therefore  $V$  should be empty (as  $U$  is not.) Therefore  $V$  is empty!  $\square$

**Remark.** As  $\mathbb{C}$  is the only non empty clopen set, we've proved that  $\mathbb{C} \subset \Omega$ , as  $U \subset \Omega$ .

**Corollary 8.** Let  $\Omega$  be a region and  $D$  be a disk in  $\Omega$ . Let  $f, g$  be holomorphic on  $\Omega$ . If  $f(z) = g(z)$  for all  $z \in D$ , then  $f(z) = g(z)$  for all  $z \in \Omega$ .

CHAPTER 4. CAUCHY THEOREM

28

Les 6

*Proof.* Let  $z_0 \in \Omega$ . Since  $\Omega$  is open, there exists  $r > 0$  such that  $\overline{D_r(z_0)} \subset \Omega$ . Let  $T \subset D_r(z_0)$  be a triangle in this disk. Since  $f_n$  are holomorphic,  $\int_T f_n(z) dz = 0$ , by Goursat's theorem. Since,  $f_n \rightarrow f$  uniformly on compact subsets, and therefore  $f_n \rightarrow f$  on  $D_r(z_0)$ , we have that!

**Definition 51.** Let  $\Omega, \Xi$  be regions such that  $\Xi \subset \Omega$ . Let  $f$  be a holomorphic function on  $\Xi$ ,  $F$  be a holomorphic function on  $\Omega$ . If  $F(z) = f(z)$  for all  $z \in \Xi$ , then  $F$  is called the analytic continuation of  $f(z)$  from  $\Xi$  to  $\Omega$ .

**Remark.** By the previous result, the analytic continuation is unique!

Les 6

**Theorem 17** (Moren). Let  $D$  be a disk, and  $f$  be a continuous function on  $D$ . If for all  $T \subset D$ ,  $\int_T f = 0$ , then  $f$  is holomorphic on  $D$ .

*Proof.* Let  $z \in D$ . Changing variables, we can assume that  $D$  is centred at the origin. Construct figure 4.4, construct  $\gamma$ . Let  $F(z) = \int_{\gamma_z} f(\zeta) d\zeta$ . Using the same method as in the proof about the existence of a primitive on a disk, we can prove that  $F(z)$  is a primitive for  $f(z)$  on  $D$ .

This means that  $F(z)$  is holomorphic on  $D$  (and  $F'(z) = f(z)$ ) This implies that  $F(z)$  is infinitely many times differentiable on  $D$ . Therefore,  $f(z)$  is differentiable on  $D$ .  $\square$

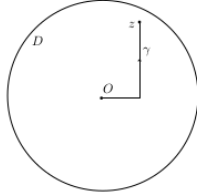


Figure 4.4: Proof of Morera.

**Theorem 18** (About sequences of holomorphic functions). Let  $\Omega \subset \mathbb{C}$  be open,  $\{f_n\}$  a sequence of holomorphic functions on  $\Omega$ . If  $f_n$  converges to  $f$  uniformly on compact subsets of  $\Omega$ , then  $f$  is holomorphic.

This result is not true in real analysis. (Every real continuous function can be approximated by polynomials? But Weierstrass exists?)

CHAPTER 4. CAUCHY THEOREM

29

Les 6

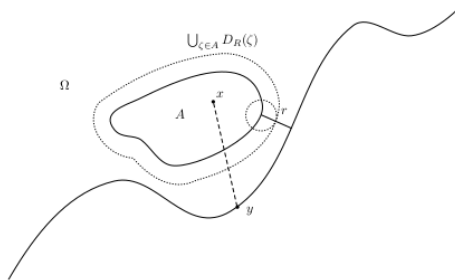


Figure 4.5: Proof of Theorem 7.

**Definition 52.** A  $\Omega \subset \mathbb{C}$  is called symmetric iff

$$z \in \Omega \Leftrightarrow \bar{z} \in \Omega.$$

Denote by

$$\Omega^+ = \Omega \cap \{z \mid \operatorname{Im} z > 0\}$$

$$\Omega^- = \Omega \cap \{z \mid \operatorname{Im} z < 0\}$$

$$I = \Omega \cap \mathbb{R}.$$

**Theorem 20** (Symmetry principle). Let  $\Omega \subset \mathbb{C}$ , open and symmetric. Let  $f^+(z)$  be a function holomorphic on  $\Omega^+$ , and continuous on  $\Omega^+ \cup I$ . Let  $f^-(z)$  be a function holomorphic on  $\Omega^-$ , and continuous on  $\Omega^- \cup I$ . If  $f^+|_I = f^-|_I$ , then

$$f(z) = \begin{cases} f^+(z) & z \in \Omega^+ \\ f^+(z) = f^-(z) & z \in I \\ f^-(z) & z \in \Omega^- \end{cases}$$

is holomorphic on  $\Omega$ .

*Proof.* It is clear that  $f(z)$  is holomorphic on  $\Omega \setminus I$ . We need to prove that  $f(z)$  is holomorphic on  $I$ . Let  $z_0 \in I$ ,  $r > 0$  such that  $D_r(z_0) \subset \Omega$ .

Let  $T$  be a triangle in  $D$  There are multiple possibilities.

- $T \subset (D \cap \Omega^+) \cup (D \cap \Omega^-)$ . Goursats handles this case.

CHAPTER 4. CAUCHY THEOREM

31

$$\int_T f_n(z) dz \rightarrow \int_T f(z) dz = 0.$$

Therefore, by Morera,  $f(z)$  is holomorphic on this disk,  $\overline{D_r(z_0)}$ . Therefore,  $f(z)$  is holomorphic on  $\Omega$ .  $\square$

**Theorem 19** (About sequence of holomorphic functions and their derivatives). Let  $\Omega \subset \mathbb{C}$  be open,  $\{f_n\}$  be a sequence of functions holomorphic on  $\Omega$ . If  $f_n \rightarrow f$  uniformly on compact subsets of  $\Omega$ , then  $\forall k \geq 0$ ,

$$f_n^{(k)} \rightarrow f^{(k)} \text{ uniformly on all compact subsets of } \Omega.$$

*Proof.* It is enough to prove the theorem only for  $k = 1$ . Let  $A \subset \Omega$  a compact subset of  $\Omega$ . Denote by  $r = \inf_{x \in A, y \in \partial \Omega} |x - y|$ . Since  $A$  is compact,  $r > 0$  (ex). Denote by  $R = \frac{r}{2}$ .

$$A \subset \bigcup_{z \in A} D_R(z) \subset \Omega.$$

Let  $F_n = f_n - f$ . By evaluation of derivatives from the previous lecture, we have

$$|F_n'(z)| \leq \frac{1}{R} \sup_{w \in C_R(z)} |F_n(w)|.$$

Therefore

$$\begin{aligned} \sup_{z \in A} |F_n'(z)| &\leq \frac{1}{R} \sup_{z \in A, w \in C_R(z)} |F_n(w)| \\ &= \frac{1}{R} \sup_{z \in \bigcup_{z \in A} D_R(z)} |F_n(z)| \xrightarrow{n \rightarrow \infty} 0. \end{aligned}$$

since  $f_n$  converges uniformly on compact subsets of  $\Omega$ , and  $\bigcup_{z \in A} \overline{D_R(z)}$  is compact. As  $R$  is fixed,

$$\sup_{z \in A} |F_n'(z)| \rightarrow 0.$$

Since  $F_n'(z) = f_n'(z) - f'(z)$ , we have that

$$f_n'(z) \rightarrow f'(z).$$

uniformly on compact subsets of  $\Omega$ .  $\square$

We can use this for power series, as this is a limit of a sequence.

<sup>1</sup>Uniform convergence implies dominated convergence

CHAPTER 4. CAUCHY THEOREM

30

Les 6

- $T \subset D \cap (\Omega^+ \cup I)$  or  $T \subset D \cap (\Omega^- \cup I)$ . Denote by  $T_z$  a smaller triangle.  $T_z$  satisfies the first case. As  $f_{T_z} \rightarrow f_T$ , since  $f(z)$  is continuous, we get that  $\int_T f = 0$ .
- Other case, split the triangle in  $T_1, T_2, T_3$ .

Now as  $f(z)$  is holomorphic on  $D_r(z_0)$ ,  $f(z)$  is holomorphic on  $\Omega$ .  $\square$

Note that we didn't really use the symmetry of the set.

Note that the analytical continuation is unique,  $f$  is the *only* holomorphic function on  $\Omega$ .

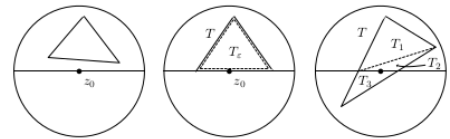


Figure 4.6: Symmetry principle

**Theorem 21** (Schwarz reflection principle). Let  $\Omega$  be an open connected symmetric set.  $f(z)$  is a holomorphic function on  $\Omega^+$ ,  $f(z)$  is continuous on  $\Omega^+ \cup I$ , and for all  $z \in \mathbb{R}$ ,  $f(z) \in \mathbb{R}$ . Then  $f(z)$  can be analytically continued on  $\Omega$ .

*Proof.* For  $z \in \Omega^-$ , define  $g(z) = \overline{f(\bar{z})}$ . Let us prove that  $g(z)$  is holomorphic on  $\Omega^-$ .

It is obvious that  $g(z)$  is continuous on  $\Omega^- \cup I$ .

Let  $z_0 \in \Omega^-$ , then  $\bar{z}_0 \in \Omega^+$ . Since  $f(z)$  is holomorphic on  $\Omega^+$ ,  $\exists r > 0, \forall z \in D_r(\bar{z}_0)$ , we have

$$f(z) = \sum_{n=0}^{\infty} a_n (z - \bar{z}_0)^n.$$

For  $w \in \Omega^-$ ,

$$\begin{aligned} g(w) &= \overline{f(\bar{w})} \\ &= \overline{\sum_{n=0}^{\infty} a_n (\bar{w} - \bar{z}_0)^n} \\ &= \sum_{n=0}^{\infty} \overline{a_n} (w - z_0)^n \end{aligned}$$

Therefore,  $g$  is analytic at  $z_0$ , therefore  $g$  is holomorphic at  $z_0$ .

CHAPTER 4. CAUCHY THEOREM

32

These lecture notes — including figures — are made while attending the lecture and have not been edited afterwards. To make note taking using LaTeX viable, I had four goals in mind:

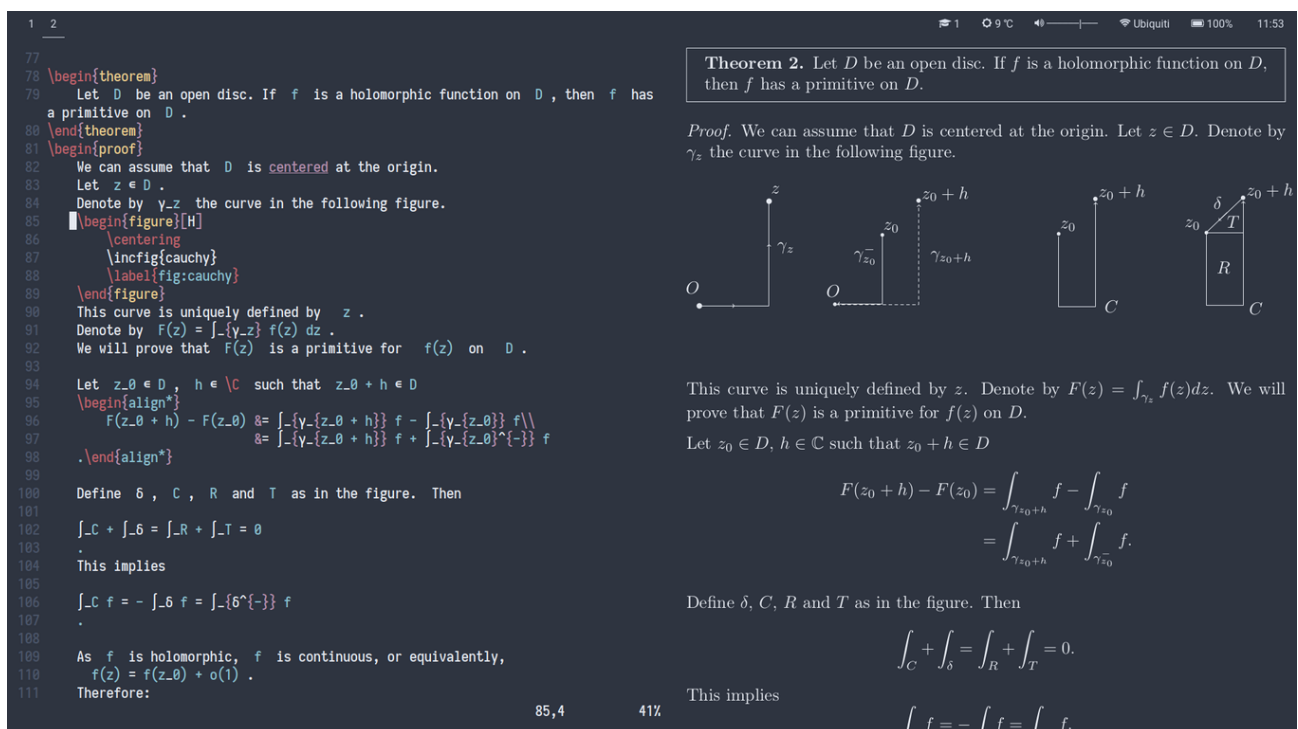
- Writing text and mathematical formulas in LaTeX should be as fast as the lecturer writing on a blackboard: no delay is acceptable.
- Drawing figures should be almost as fast as the lecturer.

- Managing notes, i.e. adding a note, compiling all my notes, compiling the last two lectures, searching in notes, etc. should be easy and quick.
- Annotating pdf documents using LaTeX should be possible for when I want to write notes alongside a pdf document.

This blog post will focus on the first item: writing LaTeX.

## Vim and LaTeX

For writing text and mathematical formulas in LaTeX, I use Vim. Vim is a powerful general purpose text editor that's very extensible. I use it for writing code, LaTeX, markdown, ... basically everything that's text-based. It has a fairly steep learning curve, but once you've got the basics down, it's hard to get back to an editor without Vim keybindings. Here's what my screen looks like when I'm editing LaTeX:

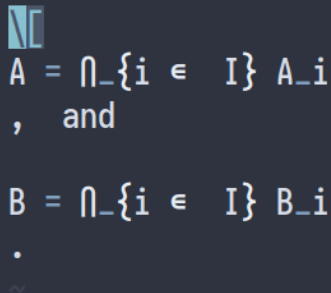


On the left you see Vim and on the right my pdf viewer, Zathura, which also has Vim-like keybindings. I'm using Ubuntu with bspwm as my window manager. The LaTeX plugin I'm using in Vim is vimtex. It provides syntax highlighting, table of contents view, synctex, etc. Using vim-plug, I configured it as follows:

```
Plug 'lervag/vimtex'
let g:tex_flavor='latex'
let g:vimtex_view_method='zathura'
let g:vimtex_quickfix_mode=0
```

```
set conceallevel=1
let g:tex_conceal='abdmg'
```

The last two lines configure the concealment. This is a feature where LaTeX code is replaced or made invisible when your cursor is not on that line. By making `\[`, `\]`, `$` invisible, they're less obtrusive which gives you a better overview of the document. This feature also replaces `\bigcap` by `∩`, `\in` by `∈` etc. The following animation should make that clear.



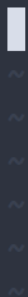
```
A = ∩_{i ∈ I} A_i
, and
B = ∩_{i ∈ I} B_i
•
~
```

With this set up, I come to the crux of this blog post: writing LaTeX as fast as the lecturer can write on the blackboard. This is where snippets come into play.

## Snippets

### What's a snippet?

A snippet is a short reusable piece of text that can be triggered by some other text. For example, when I type `sign` and press `Tab`, the word `sign` will be expanded to a signature:



```
sign
~
~
~
~
~
~
~
```

Snippets can also be dynamic: when I type `today` and press `Tab`, the word `today` will be replaced by the current date, and `box` `Tab` becomes a box that automatically grows in size.



You can even use one snippet inside another:



### Using UltiSnips to create snippets

I use the plugin UltiSnips to manage my snippets. My configuration is

```
Plug 'sirver/ultisnips'  
let g:UltiSnipsExpandTrigger = '<tab>'  
let g:UltiSnipsJumpForwardTrigger = '<tab>'  
let g:UltiSnipsJumpBackwardTrigger = '<s-tab>'
```

The code for the `sign` snippet is the following:

```
snippet sign "Signature"  
Yours sincerely,
```

Gilles Castel  
endsnippet

For dynamic snippets, you can put code between backticks `` which will be run when the snippet is expanded. Here, I've used bash to format the current date: `date + %F` .

```
snippet today "Date"
`date +%F`
endsnippet
```

You can also use Python inside a ``!p ... `` block. Have a look at the code for the `box` snippet:

```
snippet box "Box"
`!p snip.rv = '┌' + '-' * (len(t[1]) + 2) + '┐'`
| $1 |
`!p snip.rv = '└' + '-' * (len(t[1]) + 2) + '┘'`
$0
endsnippet
```

These Python code blocks will be replaced by the value of the variable `snip.rv` . Inside these blocks, you have access to the current state of the snippet, e.g. `t[1]` contains the first tab stop, `fn` the current filename, ...

## LaTeX snippets

Using snippets, writing LaTeX is a lot faster than writing it by hand. Especially some of the more complex snippets can save you a lot of time and frustration. Let's begin with some simple snippets.

### Environments

To insert an environment, all I have to do is type `beg` at the beginning of a line. Then I type the name of the environment, which is mirrored in the `\end{}` command. Pressing Tab places the cursor inside the newly created environment.



The code for this snippet is the following.

```
snippet beg "begin{} / end{}" bA
\begin{$1}
  $0
\end{$1}
endsnippet
```

The `b` means that this snippet will only be expanded at the beginning of a line and `A` stands for auto expand, which means I do not have to press `Tab` to expand the snippet. Tab stops — i.e. places you can jump to by pressing `Tab` and `Shift+Tab` — are represented by `$1`, `$2`, ... and the last one with `$0`.

### Inline and display math

Two of my most frequently used snippets are `mk` and `dm`. They're the snippets responsible for starting math mode. The first one is a snippet for inline math, the second one for displayed math.



The snippet for inline math is 'smart': it knows when to insert a space after the dollar sign. When I start typing a word directly behind the closing `$`, it adds a space. However, when I type a non-word character, it does not add a space, which would be preferred for example in the case of `$p$-value`.



The code for this snippet is the following.

```
snippet mk "Math" wA
$$ {1} $`!p
if t[2] and t[2][0] not in ['.', '?', '-', ' ']:
    snip.rv = ' '
else:
    snip.rv = ''
`$2
endsnippet
```

The `w` at the end of the first line means that this snippet will expand at word boundaries, so e.g. `hellomk` won't expand, but `hello mk` will.

The snippet for displayed math is more simple, but it also is quite handy; it makes me never forget ending equations with a period.



```
snippet dm "Math" wA
\[
$1
.\] $0
endsnippet
```

### Sub- and superscripts



Another useful snippet is one for subscripts. It changes `a1` to `a1` and `a12` to `a{12}`.



The code for this snippet uses a regular expression for its trigger. It expands the snippet when you type a character followed by a digit, which encoded by `[A-Za-z]\d`, or a character followed by `_` and two digits: `[A-Za-z]_\d\d`.

```
snippet '([A-Za-z])(\d)' "auto subscript" wrA
`!p snip.rv = match.group(1)`_`!p snip.rv = match.group(2)`
endsnippet

snippet '([A-Za-z])_\d\d' "auto subscript2" wrA
`!p snip.rv = match.group(1)`_{`!p snip.rv = match.group(2)`}
endsnippet
```

When you wrap parts of a regular expression in a group using parenthesis, e.g. `(\d\d)`, you can use them in the expansion of the snippet via `match.group(i)` in Python.

As for superscripts, I use `td`, which becomes `^{\}`. However, for squared, cubed, complement and a handful of other common ones, I use dedicated snippets such as `sr`, `cb` and `comp`.



```

snippet sr "^2" iA
^2
endsnippet

snippet cb "^3" iA
^3
endsnippet

snippet compl "complement" iA
^{\c}
endsnippet

snippet td "superscript" iA
^{\$1}\$0
endsnippet

```

## Fractions

One of my most convenient snippets is one for fractions. This makes the following expansions:

```

// → \frac{ }{ }
3/ → \frac{3}{ }
4\pi^2/ → \frac{4\pi^2}{ }
(1 + 2 + 3)/ → \frac{1 + 2 + 3}{ }
(1+(2+3)/) → (1 + \frac{2+3}{ })
(1 + (2+3))/ → \frac{1 + (2+3)}{ }

```

```

/
3
4\pi^2
(1 + 2 + 3)
(1 + (2 + 3))
(1 + (2 + 3))

```

The code for the first one is easy:

```

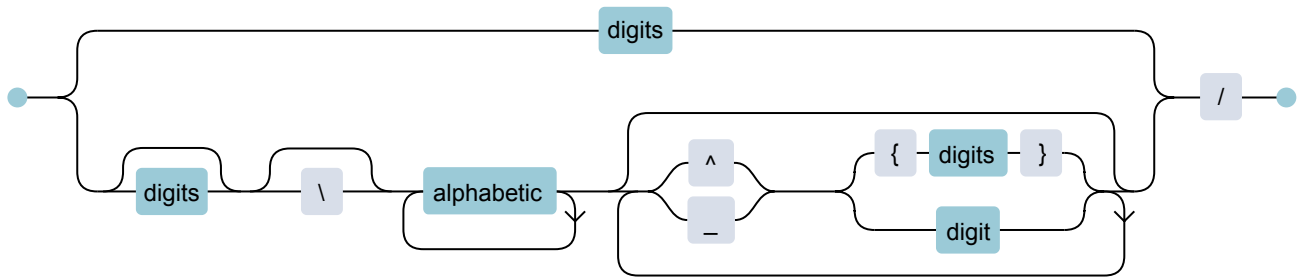
snippet // "Fraction" iA
\\frac{\$1}{\$2}\$0
endsnippet

```

The second and third examples are made possible using regular expressions to match for expressions like  $3/$ ,  $4ac/$ ,  $6\pi^2/$ ,  $a_2/$ , etc.

```
snippet '((\d+)|(\d*)(\\)?([A-Za-z]+)((\^|_)(\{\d+\}|\d))*)/' "Fraction" wrA
\\frac{`!p snip.rv = match.group(1)`}{$1}$0
endsnippet
```


As you can see, regular expressions can become quite overwhelming, but here's a diagram that should explain it:



In the fourth and fifth cases, it tries to find the matching parenthesis. As this isn't possible using the regular expression engine of UltiSnips, I resorted to using Python:

```
priority 1000
snippet '^.*\)/' "()" Fraction" wrA
`!p
stripped = match.string[:-1]
depth = 0
i = len(stripped) - 1
while True:
    if stripped[i] == ')': depth += 1
    if stripped[i] == '(': depth -= 1
    if depth == 0: break;
    i -= 1
snip.rv = stripped[0:i] + "\\frac{" + stripped[i+1:-1] + "}"
`{$1}$0
endsnippet
```

The last snippet concerning fractions I'd like to share is one that uses your selection to make a fraction. You can use it by first selecting some text, then pressing `Tab`, typing `/` and pressing `Tab` again.




$$1 + 2 + 3 + 4 + 5$$

The code makes use of the `${VISUAL}` variable that represents your selection.

```
snippet / "Fraction" iA
\\frac{${VISUAL}}{{$1}$0}
endsnippet
```

### Sympy and Mathematica

Another cool — but less used — snippet is one that uses sympy to evaluate mathematical expressions. For example: `sympy` `Tab` expands to `sympy | sympy`, and `sympy 1 + 1 sympy` `Tab` expands to `2`.



```
snippet sympy "sympy block " w
sympy $1 sympy$0
endsnippet

priority 10000
snippet 'sympy(.*?)sympy' "evaluate sympy" wr
`!p
from sympy import *
x, y, z, t = symbols('x y z t')
k, m, n = symbols('k m n', integer=True)
f, g, h = symbols('f g h', cls=Function)
```

```

init_printing()
snip.rv = eval('latex(' + match.group(1).replace('\\', '\\') \
    .replace('^', '**') \
    .replace('{', '(') \
    .replace('}', ')') + ')')
\
endsnippet

```

For the Mathematica users out there, you can do something similar:

```

|
~
~
~
~

```

```

priority 1000
snippet math "mathematica block" w
math $1 math$0
endsnippet

priority 10000
snippet 'math(.*?)math' "evaluate mathematica" wr
`!p
import subprocess
code = 'ToString[' + match.group(1) + ', TeXForm]'
snip.rv = subprocess.check_output(['wolframscript', '-code', code])
\
endsnippet

```

### Postfix snippets

Some other snippets I find worth sharing are postfix snippets. Examples of such snippets are `phat`  $\rightarrow \hat{p}$  and `zbar`  $\rightarrow \overline{z}$ . A similar snippet is a postfix vector, for example `v,.`  $\rightarrow \vec{v}$  and `v.,`  $\rightarrow \vec{v}$ . The order of `,` and `.` doesn't matter, so I can press them both at the same time. These snippets are a real time-saver, because you can type in the same order the lecturer writes on the blackboard.



Note that I can still use `bar` and `hat` prefix too, as I've added them with a lower priority. The code for those snippets is:

```
priority 10
snippet "bar" "bar" riA
\overline{${1}}$0
endsnippet

priority 100
snippet "([a-zA-Z])bar" "bar" riA
\overline{`!p snip.rv=match.group(1)`}
endsnippet
```


```
priority 10
snippet "hat" "hat" riA
\hat{${1}}$0
endsnippet

priority 100
snippet "([a-zA-Z])hat" "hat" riA
\hat{`!p snip.rv=match.group(1)`}
endsnippet
```

```
snippet "(\\(?:\w+)(,\.|\.,,)" "Vector postfix" riA
\vec{`!p snip.rv=match.group(1)`}
endsnippet
```


### Other snippets

I have about 100 other commonly used snippets. They are available [here](#). Most of them are quite simple. For example, `!>` becomes `\mapsto`, `->` becomes `\to`, etc.



```
\[
\]
```

fun becomes  $f: \mathbb{R} \rightarrow \mathbb{R} : , !> \rightarrow \mapsto , -> \rightarrow \rightarrow , cc \rightarrow \subset .$



```
\[
\]
```

lim becomes  $\lim_{n \rightarrow \infty} , \text{sum} \rightarrow \sum_{n=1}^{\infty} , \text{ooo} \rightarrow \infty$



### Course specific snippets

Beside my commonly used snippets, I also have course specific snippets. These are loaded by adding the following to my `.vimrc` :



```
set rtp+=~/current_course
```

where `current_course` is a symlink to my currently activated course (more about that in another blog post). In that folder, I have a file `~/current_course/UltiSnips/tex.snippets` in which I include course specific snippets. For example, for quantum mechanics, I have snippets for bra/ket notation.

```
<a| → \bra{a}
<q| → \bra{\psi}
|a> → \ket{a}
|q> → \ket{\psi}
<a|b> → \braket{a}{b}
```

As `\psi` is used a lot in quantum mechanics, I replace all instances of `q` in a `braket` with `\psi` when expanded.



```
snippet "<(.*)\|" "bra" riA
\bra{`!p snip.rv = match.group(1).replace('q', f'\psi').replace('f', f'\phi')`
endsnippet

snippet "\|(.*)\>" "ket" riA
\ket{`!p snip.rv = match.group(1).replace('q', f'\psi').replace('f', f'\phi')`
endsnippet

snippet "(.*)\\bra{(.*)}([^\|]*?)\>" "braket" riA
`!p snip.rv = match.group(1)`\braket{`!p snip.rv = match.group(2)`}{`!p snip.r
endsnippet
```

## Context

One thing to consider when writing these snippets is, 'will these snippets collide with usual text?' For example, according to my dictionary, there are about 72 words in English and 2000 words in Dutch that contain `sr`, which means that while I'm typing the word `disregard`, the `sr` would expand to  $^2$ , giving me `di^2egard`.

The solution to this problem is adding a *context* to snippets. Using the syntax highlighting of Vim, it can be determined whether or not UltiSnips should expand the snippet depending if you're in math or text. I came up with the following:

```
global !p
texMathZones = ['texMathZone' + x for x in ['A', 'AS', 'B', 'BS', 'C', 'CS',
'D', 'DS', 'E', 'ES', 'F', 'FS', 'G', 'GS', 'H', 'HS', 'I', 'IS', 'J', 'JS',
'K', 'KS', 'L', 'LS', 'DS', 'V', 'W', 'X', 'Y', 'Z', 'AmsA', 'AmsB', 'AmsC',
'AmsD', 'AmsE', 'AmsF', 'AmsG', 'AmsAS', 'AmsBS', 'AmsCS', 'AmsDS', 'AmsES',
'AmsFS', 'AmsGS' ]]

texIgnoreMathZones = ['texMathText']

texMathZoneIds = vim.eval('map('+str(texMathZones)+", 'hlID(v:val)')")
texIgnoreMathZoneIds = vim.eval('map('+str(texIgnoreMathZones)+", 'hlID(v:val)')")

ignore = texIgnoreMathZoneIds[0]

def math():
    synstackids = vim.eval("synstack(line('.'), col('.') - (col('.') ≥ 2 ? 1 : 0))")
    try:
        first = next(
            i for i in reversed(synstackids)
            if i in texIgnoreMathZoneIds or i in texMathZoneIds
        )
        return first != ignore
    except StopIteration:
        return False
endglobal
```

Now you can add `context "math()"` to the snippets you'd only want to expand in a mathematical context.

```
context "math()"
snippet sr "^2" iA
^2
endsnippet
```

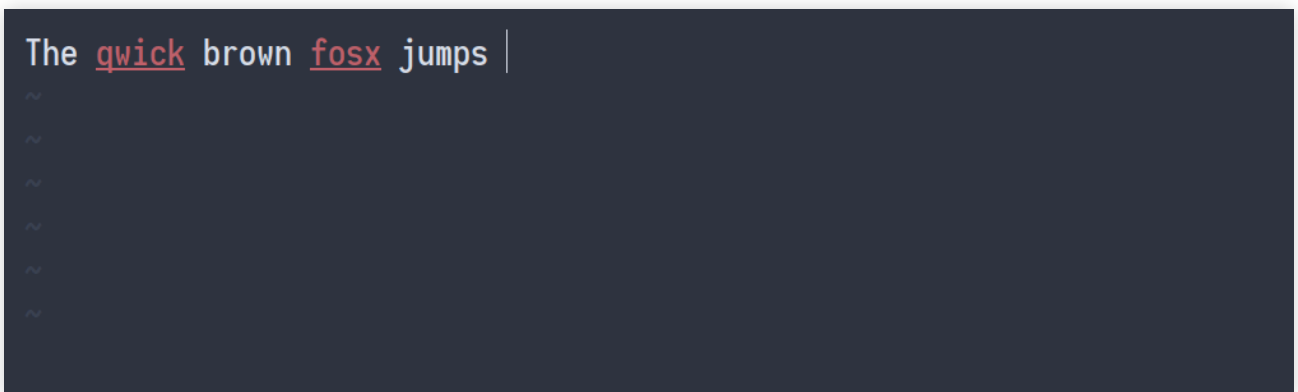
Note that a ‘mathematical context’ is a subtle thing. Sometimes you add some text inside a math environment by using `\text{...}`. In that case, you do not want snippets to expand. However, in the following case:

`\[ \text{${...$}} \]`, they *should* expand. This is why the code for the `math` context is a bit complicated. The following animation illustrates these subtleties.



### Correcting spelling mistakes on the fly

While inserting mathematics is an important part of my note-taking setup, most of the time I'm typing English. At about 80 words per minute, my typing skills are not bad, but I still make a lot of typos. This is why I added a keybinding to Vim that corrects the spelling mistakes, without interrupting my flow. When I press `Ctrl+L` while I'm typing, the previous spelling mistake is corrected. It looks like this:



My configuration for spell check is the following:

```
setlocal spell
set spelllang=nl,en_gb
```

```
inoremap <C-l> <c-g>u<Esc>[s1z=`]a<c-g>u
```

It basically jumps to the previous spelling mistake `[s` , then picks the first suggestion `1z=` , and then jumps back ``]a` . The `<c-g>u` in the middle make it possible to undo the spelling correction quickly.

## In conclusion

Using snippets in Vim, writing LaTeX is no longer an annoyance, but rather a pleasure. In combination with spell check on the fly, it allows for a comfortable mathematical note-taking setup. A few pieces are missing though, for example drawing figures digitally and embedding them in a LaTeX document. This is a topic I'd like to tackle in a future blog post.

Liked this blog post? Consider [buying me a coffee!](#)



Written by **Gilles Castel**, who lives in Belgium studying mathematics at the university of Leuven.

HOW I DRAW FIGURES FOR MY MATHEMATICAL LECTURE NOTES USING INKSCAPE →

Comments Community Privacy Policy Login ▾

Recommend 84 Tweet Share Sort by Best ▾

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS

Name

**K.Prithvi Sai** • a year ago

can you also share how you made those diagrams

27 ^ | ▾ • Reply • Share ›

**Daniel** ➔ K.Prithvi Sai • a year ago • edited

First hit on google: <https://regexper.com>

^ | ▾ • Reply • Share ›

**Brandon Wittwer** ➔ K.Prithvi Sai • a year ago

the regex diagram yeah? I'd like to know that too.

^ | ▾ • Reply • Share ›

**the-sick-big** ➔ Brandon Wittwer

• a year ago

These are called "regex railroad diagrams", a google search would bring up a list of tools you can use to make them

^ | v • Reply • Share ›

**Buwei Liao** ➔ Brandon Wittwer

• a year ago • edited

Some regex tutorial site can generate it. But sorry I can't say exactly which.

^ | v • Reply • Share ›

**killaken2000** • a year ago

Can you add an example to github so we can see the whole thing in action?

12 ^ | v 1 • Reply • Share ›

**Junix** • a year ago

Cool!!! Enormous help to me. Looking forward to note 2

"dm" is "displayed math", but what is "mk"?

6 ^ | v • Reply • Share ›

**Ciro Santilli** • a year ago

Upload your notes to GitHub.

9 ^ | v 2 • Reply • Share ›

**Marshall** • a year ago

This is wildly impressive -- sorry for the mundane question, but would you mind sharing your vim settings, e.g. color scheme and font? I've never seen vim look that good.

9 ^ | v 2 • Reply • Share ›

**fishy mcfishface** ➔ Marshall • 9 months ago

99% sure that the color scheme is Nord...

[https://github.com/arcticic...](https://github.com/arcticic)

^ | v • Reply • Share ›

This comment was deleted.

**Gilles Castel** Mod ➔ Guest • 9 months ago

Thanks for letting me know! I'll have a look at it.

^ | v • Reply • Share ›

**TEJAS SHETTY** • 9 months ago

Hats off Gilles Mind blowing

... that on Emacs, it's not working

2 ^ | v • Reply • Share ›

**Kais Jessa** • a year ago

For anyone wondering about the font, I believe it's losevka.

2 ^ | v • Reply • Share ›

**Kais Jessa** ➔ Kais Jessa • a year ago

<https://typeof.net/losevka/>

1 ^ | v • Reply • Share ›

**Bhavik** • a year ago

That's awesome!

Do you have decided under what license your snippet-configuration file is published?

Would you be ok with me incorporating (i.e. copying) some parts of your configuration into my own (publicly mirrored) dotfiles? (with attribution/link of course)

2 ^ | v • Reply • Share ›

**Rodney** • a year ago

great topic and impressive intro. Expecting drawing figures part.

2 ^ | v • Reply • Share ›

**Shourya Bansal** • 23 days ago

Whenever I try using the box, I get an error saying Unicode character ... not set up for use with LaTeX

1 ^ | v • Reply • Share ›

**I'm a lightbulb** • a month ago • edited

Thanks!

1 ^ | v • Reply • Share ›

**Masacroso** • 5 months ago • edited

Dear Gilles: I find a way to write LaTeX faster than with your method, but just in windows (unfortunately I don't find a way to do it in linux) just setting an autohotkey script (this, in continuous development). The advantage of using a script to replace keystrokes is that it doesn't depend on the TeX editor that you use (indeed you can use this script to type mathematics in forums that make use of mathjax).

1 ^ | v • Reply • Share ›

**Holger Rapp** • 7 months ago

Author of UltiSnips here. This is really strong usage of UltiSnips and gives me warm fuzzies reading about.

Thanks for writing this!

1 ^ | v • Reply • Share ›

**M B** • 8 months ago

Dear Mr. Castel,

could you please do a step by step tutorial on how to set everything up in order to be able to get most of your functionalities? I am not familiar with Vim etc. and that's why I have trouble replicating your whole setup.

I really want to take fast lecture notes with LaTeX too and you gave a great inspiration.

Have a nice day!

1 ^ | v • Reply • Share ›

**Mubtasim Fuad** ➔ M B • 6 months ago • edited

I feel it too.

I need to see a step by step video tutorial of the whole set-up.

^ | v • Reply • Share ›

**wang peter** • 10 months ago • edited

**How do you have a real-time lightning-fast rendering?**

1 ^ | v • Reply • Share ›

**Gilles Castel** Mod ➔ wang peter • 9 months ago

I just run pdf<sub>l</sub>atex on save. It is fast, but keep in mind that for these examples, I'm compiling LaTeX documents only containing one formula. Don't expect it to be that fast when working on larger documents.

^ | v • Reply • Share ›

**Martin Kunz** • 10 months ago

Awsome. I am trying to make something similar work for me. Thanks very much for the share. It will be a good inspiration.

1 ^ | v • Reply • Share ›

**Robert Pec** • a year ago

Thanks for sharing! It took several hours for me to set up this in my laptop and it turns to be very helpful in note-taking!

One thing I am curious is if it is possible to define other context besides "math()" for things like "snippets within table environment only".

1 ^ | v • Reply • Share ›

**Matteo Capucci** • a year ago • edited

Cool! I also take notes using Latex but on a completely different stack, so this is very interesting to read.

I use VS Code on Ubuntu, with just a Latex plug-in which I seldom run during notes. Contrary to you, I have very few snippets (mainly just the autocomplete suggestions) but I'm a heavy user of Latex macros, especially for course-specific notation. It saves a ton of time while typing and it helps making the code clearer and better encapsulated, too.

One thing I miss however is the ability to draw figures (aside from commutative diagrams, where tikz-cd + macros works like a charm) and I'd really like to see a post about that! For now, I Google whatever the prof is explaining and it is surprising how many identical drawings you can find.

Another one I miss is the ability to add notes/comments, especially in equations e.g. to explain what you do from one equality to the other. How did you solve this? I have an awkward command which doesn't work as well as I'd like.

Thank you for sharing your work!

1 ^ | v • Reply • Share ›

**Samuel Lampa** • a year ago • edited

Very very impressive! I did at some point take notes in LaTeX too, but using a much less hard-core approach: the InftyEditor: <http://www.inftyproject.org...> 😊 Wasn't a vim-user back then etc, which made this a natural choice for me, but it is actually surprisingly productive for writing LaTeX. A vim based solution is much more adaptable/improvable though of course.

1 ^ | v • Reply • Share ›

**major505** • a year ago • edited

Very cool. I really don't like the Latex syntax for text, so in my class of Material Resistance, I use Markdown, but I add an plugin to vs code, so I can write formulas in Latex syntax.

The problem came with drawing. I really suck in drawing in a computer, with a mouse (or worse, a touchpad).

Usually I draw by hand in a small moleskine I always carry with me, and add a number reference to the text and draw, so I could after revising the notes add a decent draw.

For graphics I usually do the same and after use



for graphics I usually do the same, and after use gnuplot or something equivalent.

The college started conflicting with work, but when I come back to college, I will probably use the new iPad air, so I can draw by hand and add to the text.

1 ^ | v • Reply • Share ›

**Andrés** • a year ago

would you mind sharing a bit more about your color scheme, font, and concealment settings?

1 ^ | v • Reply • Share ›

**Jim Lye** • 2 months ago • edited

Hi Gilles,

Thank you for these blogs!

I am entirely new to Linux and Python. It has taken me a month all by myself (and my best friend Google) to follow through your system from scratch (installed VirtualBox+Ubuntu on my PC, learned VIM, installed all the packages, guessed about your Python scripts, etc).

One question: You use "Ctrl+N" to create a new lecture note. Please tell me what is the code or script to run for that shortcut.

I am in Hong Kong. I need to find a way to buy you a coffee.

Thank you.

- Jim Lye

P.S. The most confusing part of the whole process was caused by the fact that there were two packages of the same name "yq" and I installed the wrong one in the beginning.

^ | v • Reply • Share ›

**Jim Lye** ➔ Jim Lye • 2 months ago • edited

Oh I have figured it out:

```
#!/usr/bin/python3
from courses import Courses
lectures = Courses().current.lectures
new_lecture = lectures.new_lecture()
new_lecture.edit()
```

This has become a self teaching Python course.  
=)

^ | v • Reply • Share ›

**Adhyyan Narang** • 2 months ago • edited

Hi!

- For vimtex the default completion command `ctrl+X`  
`ctrl+O` seems very long. Do you use something else instead?

- I faced the issue with the conceal as another person below. But `hi Conceal ctermbg=none` was unable to fix the issue. I am using `iterm2` on macOS.

^ | v • Reply • Share ›

**Fraser** ➔ Adhyyan Narang • 24 days ago

if you set the colorscheme after you set highlighting in the vimdc file, the highlight will be overwritten

^ | v • Reply • Share ›

**BrunoAnt** • 2 months ago

Wow, I'm very happy by founding this blog.

I wanted to do notes with LaTeX for a while now on. And you will make it possible!

Thanks for posting this kind of work, I hope you keep it going.

You are amazing Gilles!

Best wishes from México!!

^ | v • Reply • Share ›

**BrunoAnt** ➔ BrunoAnt • 2 months ago

I was wondering if there is an alternative to do all of this in a tablet (i know it depends pretty much on the kind of tablet you have). Im trying to take notes just like you are but in a tablet. It will be reaaaaally interesting if you have a chance to explore this!

^ | v • Reply • Share ›

**Simon Mrsle** • 4 months ago

Hi ! First, this work is amazing, thanks ! You'll drink some coffees ahah ^^.

I don't know why, but when i install the base16-nord theme, the colors

are reversed in my tex file, to some extent. While your `"\begin{"` are

red and yellow, mine are blue. And my maths content is in red.

Does someone have the same "issue" ?

 View — uploads.disquscdn.com

^ | v • Reply • Share ›

**Gilles Castel** Mod ➔ Simon Mrsle • 3 months ago

After a skype session, we found out the problem was resolved by manually applying the following pull request (which was never merged):  
<https://github.com/dylanara...> You can do this by executing the following command:

```
echo '{"special":  
{"background": "#2E3440", "foreground": "#D8DEE9",  
{"color0": "#3B4252", "color1": "#BF616A", "color2":  
> `locate base16-nord.json`
```

^ | v • Reply • Share ›

**Gilles Castel** Mod ➔ Simon Mrsle  
• 4 months ago • edited

Thanks a lot! You're not the only one. You can find the answer [here]. If you have any problems, feel free to ask away!

^ | v • Reply • Share ›

**Simon Mrsle** ➔ Gilles Castel  
• 4 months ago

Thanks for your answer, but unfortunately i didn't find the solution... Your link leads to this discussion, but i can't see anybody with the same issue...

^ | v • Reply • Share ›

**Gilles Castel** Mod ➔ Simon Mrsle  
• 4 months ago

Hmm, that's weird, I thought it linked to my reply to Michael. By searching for "there is no red highlighting of commands", you should probably be able find his comment.

^ | v • Reply • Share ›

**Simon Mrsle** ➔ Gilles Castel  
• 4 months ago

I'm going crazy, it seems like it's impossible to make it work...  
Which terminal are you using ?

^ | v • Reply • Share ›

**Simon Mrsle** ➔ Gilles Castel  
• 4 months ago

After some more tests, it seems that the colorscheme is not loaded

at all and that i only have the colors of my terminal, but not adapted to vim.

Indeed, whether i start vim with the wal.vim found or not, it doesn't change anything...

^ | v • Reply • Share ›

**Simon Mrsle** ➔ Gilles Castel  
• 4 months ago • edited

Maybe the problem is the way i "install" wal.vim... If i don't put it manually in the ~/.vim/colors directory, it's not found.

I installed it with git clone in my ~/.vim/plugged directory

^ | v • Reply • Share ›

**Simon Mrsle** ➔ Gilles Castel  
• 4 months ago

Oh yes i missed it sorry. Actually, i already did what you mentioned, but the problem persists. I built my vimrc with the same config you mentioned on your github. It's strange, cause the colors seem great in my terminal... But even with a minimal vimrc, it doesn't work... The plugins are well installed, and i put the wal colorscheme in my ~/.vim/colors directory. It is found and applied.

```
set nocompatible
syntax on
set number
filetype plugin indent on
```

© 2020 All rights reserved.