

# GM's World of Wonderful Words

G. M. Seskii

## Contents

<b>Contents</b>	<b>1</b>
<b>1 Fear Me</b>	<b>1</b>
1.1 You Aren't Listening . . . . .	1
1.1.1 That's it, Time to get Eaten . . . . .	1
<b>2 "Basic" L<sup>A</sup>T<sub>E</sub>X stuff</b>	<b>2</b>
2.1 From the "Short" guide . . . . .	2
2.1.1 It's really not short at all... . . . . .	2
<b>3 PACKAGES!</b>	<b>5</b>
3.1 Too many packages... . . . .	5
3.1.1 Someone get Kirara to get rid of these . . . . .	5

Oh boy oh boy what have we here a fun little day on the word world? How interesting, let me just fill this document with all manner of text so we can figure out exactly how the default paragraph will appear. Good. Neat, even. Perhaps funky like chicken!

## 1 Fear Me

### 1.1 You Aren't Listening

#### 1.1.1 That's it, Time to get Eaten

Anyway we appear to be using TeXworks Editor, which is neat and helpful already. Let's see what features it has by default. "Running" the file will generate it rather quickly. Default is pdfL<sup>A</sup>T<sub>E</sub>X which makes a pdf. Which is nice. There are several others.

We have some templates we could use, as well as, apparently, some minor form of version control, reverting to saved. There's a citation helper (if I had that) and a whole ton of preferences, which are mostly visual and technical things that probably don't need to be adjusted, like the icon size, document previous properties, the like.

Shift-tab doesn't work, so use the ctrl+[ ctrl+] commands. There's also scripting functionality, but we probably won't need to use that. And, oh, a link

to the short manual for this program, hmm, that's probably something to look into. Yep. ((It wasn't all that good, honestly...))

$$9 - 9 - 9 = -9$$

Anyway, we will now work through <https://mirrors.rit.edu/CTAN/info/lshort/english/lshort.pdf> and record what we find. After that we will investigate packages.

## 2 “Basic” L<sup>A</sup>T<sub>E</sub>X stuff

### 2.1 From the “Short” guide

#### 2.1.1 It's really not short at all...

A list of awkward characters typed by adding a slash to them: # \$ % ^ & \_ { } ~ . You can't do this with the backslash, you have to use the `textbackslash` command to get `\`. I was curious what all of them did so I looked it up. Other slash / is fine, though the slash command should be used to allow for hyphenation.

`\` is obviously the “command” character. `{ }` are there to contain things. `%` is for comments. `^` is evidentially reserved for exponent in math mode, and it won't print anything in normal mode. (Just throws an error.) This is the same for `_`, the subscript. I know `$` as the math sign. `&` is called the tabulator (tables) and `#` is for parameters (which I've never seen used). `~` is the protected space.

Fun fact: you can add hats with the commands! If there's no slash after, letters are fun: `~` `â` `â`. No need for math mode! Notably, of these `~` is the only one that doesn't break when typed normally, as it converts to protected space. (Technically `%` too, I guess.)

The math command “sim” gives a proper tilde. `~`

...

Sometimes we need to box things to make sure they stay together. `mbox` does this without making the box. `fbox`, meanwhile... are you not entertained!?

Remember to use ``` for beginning quotes. “this” is not like “this”. The `enquote` command can also be used.

Dash-dash-dash—dash—dash!

$\emptyset\tau$

Ah yes, the importance of references. First, use the `label` command to label something. Now let's refer back to that something using `ref 2.1.1` and `pageref 2`. Note that these are not clickable and just point to where the reference is<sup>1</sup>. ((A package enabled later made the references clickable)).

Now there are times we want to set up specific environments via the `begin` and `end` commands, each given an argument for said environment. I'm familiar with the `matrix`, but it can also be used for lists with “`itemize`” and “`enumerate`” while “`description`” is... descriptions. There are lots of these. There's also ones

---

<sup>1</sup>also here's the footnote command

for text alignment, “center” is pretty helpful. Also special quotes, verse... an “abstract” ... (look at me mixing dots, how scandalous). “verbatim” can be used to break the rules.

##%\$#@#%# Look at how bad I am!  
 Typewriters are fun and cool!  
 Cirno vs Paimon go

Lots of fun symbols can be typed in directly: €%ŽœŒÜ?WÛ. Not all, though, Unicode is not enabled by default, and we shouldn’t need it in the first place. Some characters also seem to trip it up based on what “compiler” (or engine) we are using.

The “tabular” environment does tables.

Anemo	Geo	Electro	Dendro	Hydro	Pyro	Cryo	Quantum	Imaginary
Wind	Rock	Lightning	Nature	Water	Fire	Ice	Dark	Light
Teal	Yellow	Purple	Green	Blue	Orange	Cyan	Blurple	Yellow but different

Hmm, dunno how to justify that away from the margins yet. Table tooooo wide atm. Maybe fix...

There is also a figure environment. This gives us the ability to do... GRAPHICS! ...using a package graphix so that’s for later. Regardless, note that both the figure and table environments are considered “floating” environments and may move around all over the place.

Floating environments (floats) can have specified arguments of where to place them (h)ere (t)op (b)ottom (p)age and (!). The last one is just SHUT UP PLACE THE STUPID DIAGRAM ALREADY! default is tpb.

Oh I suppose we have the math section here. I kind of already know how to use that, but let’s be sure. Let’s see what we can and can’t do with loading amsmath. We already know we can do basic incline  $x - x$  math and complex set apart

$$x^2 - x_2\spadesuit$$

math, but what else is on offer? One thing we have is the equation environment, which will automatically number things.

$$1 = \sum_{n=1}^{\infty} \frac{1}{2^n} \tag{1}$$

Look at that nice label. Pristine. It does appear that amsmath may be automatically enabled... AHA! It is not! Had to enable it to use the text command in math. Interesting...

NOTE: apparently it is bad practice to use \$\$...\$\$\$. Use the brackets instead.

Handful of things I did not know how to do in MATH mode. The surd command gives  $\sqrt{\phantom{x}}$ , which is sometimes helpful for conciseness. Overline can produce  $0.\overline{0333}$  which is neat, not to mention the excessive underlining abilities!

$$\frac{111111}{\underline{\underline{\underline{\hspace{1cm}}}}}$$

Which does not look like but you know what that's still cool. There's also the overbrace and underbrace.

$$\overbrace{\text{Nahida}}^{\text{cage}} \\ \underbrace{\hspace{1cm}}_{\text{birdness}}$$

Hmm, that brace isn't very smooth. Wonder what can be done about that, if anything? Also, the way to get limit notation on things that aren't the lim command is to define a new math operator. Also the qquad command can give a nice break. The binom command is for binomials. Now for something really neat: the stackrel command!

$$\stackrel{8}{\stackrel{88}{88}}$$

Make STACKS!

There's also stuff for dealing with long equations but that's irrelevant to me right now.

Arrays and matrices are basically tables. An array has no bounding unless you provide it. A matrix also does not, but a bmatrix has the  $\begin{bmatrix}$  bounds. In fact "b" is just a letter code stucn on the front: we have  $\text{p() B\{ } v\|}$  and  $V\| \|$

In math, there are spacing commands:  $\, , \: , \; , \quad$  space quad qquad. With space being " ", y'know, empty. The  $\! \!$  command produces negative space! The phantom command reserves space for a character that is not shown. How mysterious that space there is. Almost like it's the size of MW...

The math scripts are mathcal, mathfrak, and mathbb. There's also  $\text{Re } \Re$  and  $\text{Im } \Im$  for your real and imaginary needs. Also  $\aleph$  and other random symbols exist.

$A\vdash B[C]D[\mathcal{E}]\mathcal{F}\{G\}\mathcal{H}(I)\mathcal{J}|\mathcal{K}\|\mathcal{L}\updownarrow\mathcal{M}\updownarrow\mathcal{N}\backslash\mathcal{O}\mathcal{P}\sqrt{Q}\mathcal{R}\nabla\mathcal{S}\mathcal{T}\sqcup\mathcal{U}\cap\mathcal{V}\sqsubseteq\mathcal{W}\supseteq\mathcal{X}\S\mathcal{Y}\dagger\mathcal{Z}\ddagger$

$\mathfrak{A}\mathfrak{a}\mathfrak{B}\mathfrak{b}\mathfrak{C}\mathfrak{c}\mathfrak{D}\mathfrak{d}\mathfrak{E}\mathfrak{e}\mathfrak{F}\mathfrak{f}\mathfrak{G}\mathfrak{g}\mathfrak{H}\mathfrak{h}\mathfrak{I}\mathfrak{i}\mathfrak{J}\mathfrak{j}\mathfrak{K}\mathfrak{k}\mathfrak{L}\mathfrak{l}\mathfrak{M}\mathfrak{m}\mathfrak{N}\mathfrak{n}\mathfrak{O}\mathfrak{o}\mathfrak{P}\mathfrak{p}\mathfrak{Q}\mathfrak{q}\mathfrak{R}\mathfrak{r}\mathfrak{S}\mathfrak{s}\mathfrak{T}\mathfrak{t}\mathfrak{U}\mathfrak{u}\mathfrak{V}\mathfrak{v}\mathfrak{W}\mathfrak{w}\mathfrak{X}\mathfrak{x}\mathfrak{Y}\mathfrak{y}\mathfrak{Z}\mathfrak{z}$

$A\oslash B C D E F G H \sim I J K \neg L < M > N \ltimes O \rtimes P | Q | R \setminus S \sim T \approx U \cong V \gtrsim W \lesssim X \frown Y \rightharpoonup Z F$

You might notice that there's a lot of random symbols here. That's the lowercase letters, they've been transformed into other symbols, presumably since they don't actually exist in the script we want.

The mathbf command gives you boldface. Boldmath works too but only outside of math mode, it holds math mode within it. boldsymbol can come from amsby or bm packages.

Alright, now, bibliographical stuff. Command  $\text{bibitem[label]\{marker\}}$ . The marker is what we use to refer to it when we use the cite command. The label is just the number by default, "source [1]" sort of deal. The actual citation goes directly after the bibitem command.

Size comands are a little strange. Instead of the usual  $\text{command\{ } layout}$ , it's  $\{ \text{command text} \}$  layout. We shall now allow the sizes to be self demonstrating:

tiny scriptsize footnotesize small normal size large Large LARGE huge  
Huge

Evidentially there is not HUGE. There are also the basic font adjustment commands. Which are not done in the above manner.

`textrm texttt textmd textup textsl emph textsf textbf textit TEXTSC textnormal`

The `math` commands can do text as well. `mathAA`, where `AA` is some code. `rm` is roman, `bf` is bold, `sf` is sans serif, `tt` is typewriter, `it` is italic, `cal` is caligraphic (breaking the pattern I see), and there's also a "normal" but it's an odd day when you need to specify that.

The `par` command is equal to a blank line. The `PARagraph`. `Geddit?` `noin-`dent removes the indent. `Indent` adds it. Now the stretch command can be used to make sure things fill the line:

M V E M J S U N

There are also spacing commands. Which I should basically never need. Also messing with the `*page layout*` oh that's always fun.

There is a standard picture environment. We should never use it, and instead rely on the `TikZ` package. Or, maybe, just maybe, do the graphs in python and export them...?

There are also customization options where we can define new operators, new commands... but seriously the packages should be able to cover it.

The `rule` command can be used to draw boxes and lines.

Some notes on "compilers": there are several built into TeXworks. The default was PDFLaTeX, but some suggestions indicate that XeLaTeX is better. XeLaTeX does assume unicode and apparently that's convenient. PDFLaTeX has more sophisticated microtypography. So... take a pick? At this point the two have minimal difference in final output result. Babel appears to have issues. There's also LuaLaTeX.

Anyway that's supposedly the "basics" but we did have to dig into a few packages. Let's go dig into a few more!

Hey people, here's a margin note using the `marginpar` command! Neat, innit?

## 3 PACKAGES!

### 3.1 Too many packages...

#### 3.1.1 Someone get Kirara to get rid of these

I will attempt to restrict this to ones I intend to use, or at the very least find fun.

**amslatex:** is by far the most important of the packages, as it's how all the math is DONE. (Or, well, most of it, anyway; basically indispensable though). The full `amslatex` package contains `amsmath` and `amscls`. `amscls` is not relevant to us as it's mainly for document preparation for the American Mathematical Society. `amsmath`, though, contains a truly tremendous amount of THINGS it can do. Fortunately I actually mostly know how to use it already due to

experience, neat! And we demonstrated its capabilities up there a bit. Just make sure to load `amssymb` to make sure you get the full symbol package. Also, there is `mathtools`, which expands `amsmath`. `Mathtools` is a collection of various expansions, and if you load it it loads `amsmath` automatically, and can even pass arguments to it. It is mostly just bug fixes and small features to let weird math be written down.

**biblatex:** COMPREHENSIVE BIBLIOGRAPHY! This one's important for obvious reasons. Let's see what it can do! Oh boy the documentation is several hundred pages long whoooof... most of the work for `biblatex` happens outside the main document. You create a `.bib` file to hold all possible citation formats, and then load it in the preamble. `Biblatex` requires `csquotes`. Anyway, once we set up the `.bib` file and load it, we can just cite things with the `cite` command, using the name we gave the source; Mordecai 2020. Then we can generate the bibliography with `printbibliography`.

## References

- McGee, Nameless (1111). "Does the Abyss Stare Back?" In: *Abyssal Investigations* 11.1, p. 111.
- Mordecai, Mortimer (2020). *MORT*. 1st ed. Beyond the Veil: Those Beyond the Veil. 666 pp.

NOTE: To properly generate this, build LaTeX, then build `biber`, then build LaTeX again. Probably something to do with file order. `Biblatex` has FAR more resources than I need, with almost every kind of citation imaginable.

**fancyhdr:** fancy headers. Adjusting the specificity of what is on the headers and footers, quite important. By default this document doesn't even have them. This appears to be primarily done by just inserting the type of header and/or footer. In order to enable this we first have to set `pagestyle` to `fancy`, and then clear the headers by calling `fancyhead` with an empty call. All of the header commands are done like `command[location]{content}`, where the location is given by various letter codes. O and E are for odd or even, LCR are directional, and HF are sometimes implied but can mean header and footer. Neat!

**xcolor:** `COLORS`. Nice how self-demonstrating this entry is, eh? simply use the `color` command to color everything after it. Remember to switch back to black at the end! There are plenty of named colors, and more colors can be accessed by setting the options for `xcolor`. This also implements `colorbox` for background coloring/highlights. Custom colors can be done with the `definecolor` command, which has `rgb`, `cmyk`, and `gray` options for definition. What fun! `pagecolor` can also be set, but that would be a bit crazy, huh? A proper highlight would require the `lua-ul` package, but chances of that being necessary are low.

**hyperref:** hyperlinks, for both internal and external linking. In fact, it should be able to handle a table of contents automatically. Generating a table of

contents proves that it can, and amazingly it also connects with the bibliography! So anything that has a reference can be clicked and it'll be arrived at. Amazing how that kind of handles itself, but it means I have to be dilligent with my references! However they do appear as ugly red and green boxes... aha, one can either hide the links by making them all invisible, OR by setting up hyperref with xcolor. External links are also allowed with the href and url commands: [A Special Link](https://www.fimfiction.net/user/275276/GMSeskii), <https://www.fimfiction.net/user/275276/GMSeskii>. Local file links are also supported, as are direct links from anywhere if you want to manually set them with hypertarget and hyperlink.

**graphicx**: for graphics and figures. Yes we need to put images into our pdfs thank you. Import graphicx, then set the graphicspath to the necessary path to the images. Then just use the command includegraphics. However, that doesn't try to do anything with the image size or caption it or anything. Sooooo you can pass arguments, such as scale= or angle=. Figures can also be labeled but that's a normal thing, not really a function of graphicx itself.



Figure 1: Ever so *slightly* out of alignment...

**adjustbox**: gain precise control over any boxed content, such as tables. Graphics can be adjusted with graphicx, but what of our fancy tables? Well, let's try it, our previous table was a bit too large.

Anemo	Geo	Electro	Dendro	Hydro	Pyro	Cryo	Quantum	Imaginary
Wind	Rock	Lightning	Nature	Water	Fire	Ice	Dark	Light
Teal	Yellow	Purple	Green	Blue	Orange	Cyan	Blurple	Yellow but different

Success! Give the adjustbox a box and it'll do stuff to it. Neat! Has similar controls to the figure environment. Can also frame the box, if you need that, as well as a ton of other options.

**siunitx**: Formatting units properly. Also has functionality for making tables line up on the decimal point. Lots of other number display commands, can even display computer exponent 1e5 numbers expanded  $1 \times 10^5$ . The post powerful and readily usable command is probably qty, which combines numbers and units. Say, 1 943 678 m s<sup>-1</sup>. Note the subtle spaces in the large number. Neat! Of course we can set the per-mode to fraction and get 1 943 678  $\frac{\text{m}}{\text{s}}$ . The prefixes can also be added with their own commands. Also wow it goes from -30 to 30 in terms of the orders of magnitude, whew.

**mdframed:** Do you want more involved, complex, and page-breakable boxes? Well look no further! Behold this self demonstrating amazingness!

**bm:** Boldify math. That's it. The command is `bm`.  $9\pi N, 9\pi N$

**csquotes:** Hey, getting tired of manually typing in opening and closing quotes? Well look no further! `csquotes` "solves" the problem! Was also required for bibliography stuff but that's not the point I'm using it for. We are now freeeeee from textural idiosyncrocies. Which I totally spelled correctly, yes. Anyway this package can actually do a lot of contextual and nested proper quotes, may be worth looking into if quotes are becoming a problem

**cmap:** make your resulting pdfs searchable and copyable, just to be convenient. Very important to include, but doesn't need anything done besides being enabled.

**wasysym:** a font that has a ton of symbols in it, such as the astronomical symbols. Some math symbols too,  $\square$ . Then we have  $\otimes \clubsuit \odot \oplus \sim \infty \approx \text{musical notes}$  among others.

**mhchem:** You might think it odd that this is here and not the physics package. However, the physics package didn't really provide anything new that I couldn't do already, aside from bra-ket notation, and I won't be using that. However I will need to do chemical things every now and then (we love atmospheric chemistry). So... test!  $\text{H}_2\text{O}$  and  $\text{H}_2\text{O}_4\text{U}$  and  ${}^{111}_{11}\text{Th}_9^+$  yep it all checks out.

**cancel:** draw lines through things.  $\cancel{9}$  things  $\cancel{N}$   $\cancel{\infty}$  and perhaps the most useful,  $\cancel{\frac{9}{0}}$

**circledsteps:** numbers with circles around them. Good for annotating things.  $\textcircled{9}$   $\textcircled{3.14159}$  hmm the larger circle appears to be screwing up. Hmm. Oh well, single numbers still work fine, this is mainly for highlighting n stuff.

**csvsimple:** work with csv files. Might be neat to make an automatically updating table at somepoint. Would also greatly simplify making of tables if it can just be autoloaded from csv.

**fontawesome:** adding symbols via a font. Similar to `wasysym`, just providing more symbols.  $\textcircled{5}$  . Focused on lots of web items.

**longtable:** for when your table is loooooooooong. Yeah just lets you make multipage tables. Notably fails in the double column environment, but if you have a long table you probably want it to straddle anyway. (A puzzle for a later day...)

**fp:** If you ever need to do floating point arithmetic for some reason. Probably best used for calculating table sizes or spacing, rather than displaying numbers you could just type. The major weirdness in this package is that it doesn't accept normal arguments in braces, but rather "variables" which are defined with slashes. `\FPset\x\y` sets x to y. If a specific value is desired, then it is set with braces. `\FPset\x{2}`. There are arithmetic results and the like, and the `FPprint` value. Allow me to print x: 2. There's also quite a few odd functions and logic that can be used. Also pi. 3.141592653589793238



**At the end here I will note packages I don't intend to use, but are still important and may be needed at some point.**

**babel:** managing multiple languages and language support in general.

**TikZ:** drawing your own graphics bit by bit. But why would we bother when we could just include images from tools better suited to this? Even the flowchart functionality can be done by Mermaid Chart! The pstricks package is very similar in being probably unnecessary.

**booktabs:** Gain more control over tables... if needed, strictly speaking it's probably not necessary unless you're doing something odd or really want it to look nice. See also nicematrix for the math version. And colortbl for the colorful version.

**cleveref:** cross-reference features improved! Always more references! Seems redundant with hyperref.

**microtype:** when you *really* want control over your document. Lots of tiny adjustments, it seems.

**multicol:** For messing with text columns in the document. Normally we can get away with just two columns, but sometimes this might be necessary, especially if we end up with some ugly columns that need tweaking.

**geometry:** at important package that allows you to adjust margins, paper size, etc. But we shouldn't use it unless we need to.

**parskip:** Option to not indent your paragraphs and just make them separated. Affects entire document though, so not good for doing individual sections, but would be good if you wanted to use non-indent format for whatever reason.

**physics2:** so the usual package for doing this was physics, but physics apparently causes problems and conflicts, so physics2 is here. physics2 is big, so individual modules have to be loaded with usephysicsmodule{modules}. The ab module does automatic bracing. Contains quantum bra ket notation in ab.braket and braket modules, but are incompatible with one another. Quickly realized that this doesn't actually provide anything I can't already do, so... to the bottom of the doc with you!

**caption and subcaption:** control over captions and subfigure caption support. And stuff. May be useful one day, but normal figure caption functionality seems fine for now.

**iftex:** identify which "compiler" (engine) you're using, add some conditionals for it.

**wallpaper:** background images! Probably won't ever need to use this but you never know.

**section:** control over section headings and styles. Not necessary until it is, such as a formatting requirement for submission.

**subfig:** for when you need figures in your figures. When will it be necessary? I dunno.

**censor:** Somewhat Silly. Adds the ability to black out information. Or any color.

**bookshelf:** Somewhat Silly. Makes a bookshelf out of your bibliography.

**logicpuzzle:** Somewhat Silly. Set up logic puzzles. However, the nice colors and grids here might be useful for other things, actually...

**halloweenmath:** SILLY! Use halloweeny symbols in your mathematics, including some pretty neat and complicated ones. Skulls, pumpkins, ghosts, you name it!

**chickenize:** SILLY! The silliest of packages! Rainbow text, turn things into chickens! CHICKENS! Requires LuaLaTeX though.