



Efficient note-taking in  
Emacs with howm package █

# TABLE OF CONTENTS

Introduction .....	2
The Concept of Howm .....	3
Emacs Key Combinations .....	5
Installing Howm .....	8
Howm Main Menu .....	11
Creating Notes .....	15
Working with Lists .....	18
Sorting Notes in Lists .....	24
Working with Links .....	27
Keywords in Howm .....	33
Search Functions .....	38
Filter Functions .....	41
Deleting Notes .....	44
Stitching Notes .....	46
Handling Long Notes .....	49
Working with Buffers .....	53
Quick Notes .....	55
Duplication .....	57
Descriptive Names .....	58
Task Management .....	59
Advanced Settings .....	67
Questions and Concerns .....	74
The List of Main Howm Commands .....	75

# INTRODUCTION

This is a tutorial on taking notes in Emacs using the `howm` package. I assume that readers have already recognized the importance of “thinking by writing” and are familiar with the fundamental principles of knowledge management.

Essentially, you can count these principles on one hand:

1. Take notes on anything you find useful, important, interesting.
2. Limit the length of each note to one idea; this makes it easier to construct them later on in any manner.
3. Link notes when their respective ideas engage in a dialogue.
4. Use a reliable program that you can trust for your writing.

This tutorial is about such a program. I will try to show why functions from the `howm` arsenal are important for organizing intellectual work. However, I do not want to impose `howm` on the reader. There are other good solutions (even within Emacs). I only hope that you will appreciate the simplicity and reliability of this package.

This text is tailored for absolute beginners in Emacs who are just learning the basics. Throughout the narrative, I will explain some universal techniques that are applicable not only in `howm`. Naturally, experienced users may get bored with all this. They are free to choose the sections of the tutorial that best suit their needs.

I did not intend to create an exhaustive technical manual, so I have omitted some specifics. My goal is to showcase the capabilities of `howm` as a note-taking system. It’s important to mention that this is my subjective perception of the tool. This book is not about how one must use `howm`, but rather explores how it can be effectively utilized.

While writing this text, I repeatedly pestered the creator of the `howm` package – Kazuyuki Hiraoka – with questions. I would like to express my gratitude to him for his patience and willingness to help. Domo arigato!

The text is distributed under the GNU Free Documentation License and is available for free use.

Andrei Sukhovskii, 2023

# THE CONCEPT OF HOWM

The name “howm” is an acronym for the Japanese phrase or slogan “Hitori Otegaru Wiki Modoki” (一人お手軽 Wiki もどき), which can be roughly translated as “Handy Own Wiki Mimicry.”

The howm package surprises with its compactness and elegant solutions. It makes sense that the program is from Japan, the land of “Wabi-sabi.” The author, Kazuyuki Hiraoka, is a professor at Asia University (Japan), Faculty of Business Administration, Department of Data Science. The project is over 20 years old, as Hiraoka started it in 2002.

What’s the appeal of howm? What’s so special about it? In a nutshell, it was invented for unorganized people.

Hiraoka made a good point about the dilemma of note-taking:

If you don’t organize notes, it’s wearisome to read them.

If you try to organize notes, it’s wearisome to write them.

The need of trade-off is worrisome.

“Wearisome” is an important keyword. There is a big difference between “I can do it” and “I can do it easily.”

And then he writes:

The point of compromise may vary from person to person, but I prioritize the ease of writing. I have designed a memo tool to create an environment where one can freely jot down notes without feeling pressure to “organize” them and still maintain coherence.

Hiraoka claims that rigid structure and categorization quickly become a hindrance. Howm’s motto – “Write fragmentarily and read collectively” – in my opinion, captures the essence of a good note-taking system.

Howm is suitable for working with any text files<sup>1</sup>. It doesn’t matter whether it’s in txt, Org, or Markdown format; by placing the files in the howm folder, you incorporate them into the knowledge base.

Each new note in howm is a new file that gets a unique number based on the date and time it was created. There are no tags or even links in the traditional sense. Instead of this, Kazuyuki Hiraoka employed a clever trick. In the howm, a link serves as a trigger for a search query. The outcome is a

---

<sup>1</sup> Howm is a minor mode in Emacs that can befriend many other modes.

list that contains all the notes where the link appears (file name, title, etc.)<sup>2</sup>

The functionality of howm was inspired by the file organization system of Japanese economist Yukio Noguchi.

In a nutshell, Noguchi's system suggests replacing thematic information distribution with a frequency-based distribution, ensuring that the most important documents are always at hand.

Howm follows a similar principle. It does away with categories or sections<sup>3</sup>, opting for a general list where notes are organized based on the date of the last edit. This algorithm of working with information turns out to be more efficient than striving for perfect order.

Howm's architecture drew inspiration from various applications, including ChangeLogMemo, Q-Pocket, HyperCard, HashedWiki, and more.

The syntax of howm may appear unusual initially. Nowadays, the public is used to the markdown format and expects to encounter it everywhere. In defense of howm's syntax, it predates this widespread adoption.

We will delve into a discussion of howm features on page 11, but before that, I need to make a couple of introductory remarks about using Emacs, installing new packages, and so on.

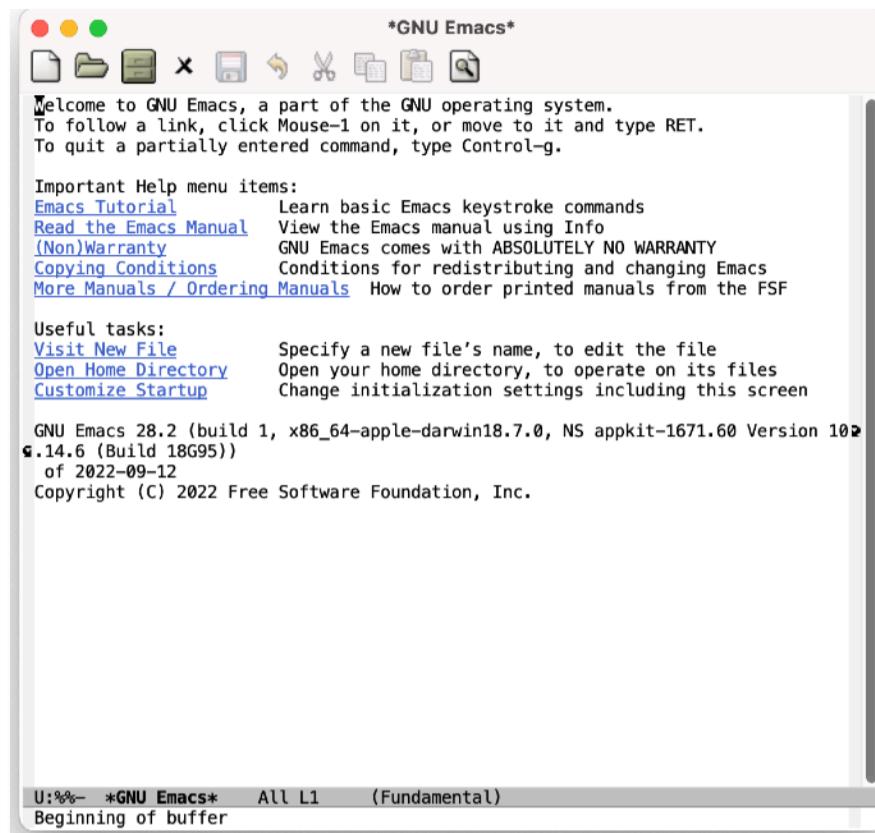
---

<sup>2</sup> Similarly, links handling was organized in Notational Velocity, an application developed by Zachary Schneirov in 2002.

<sup>3</sup> This approach is also common among zettelkästen adherents.

# EMACS KEY COMBINATIONS

I'm assuming you already have Emacs installed on your computer.



Congratulations! You can work with documents, create and edit notes, track various tasks, and do many other things.

In the next section, we'll discuss the installation process for howm. However, before delving into that, I need to cover basic keyboard commands in Emacs.

If you are new to Emacs, many things may seem unfamiliar. The best advice is to go through the built-in Emacs Tutorial, which is readily available on the start screen right after launching the program (first hyperlink).

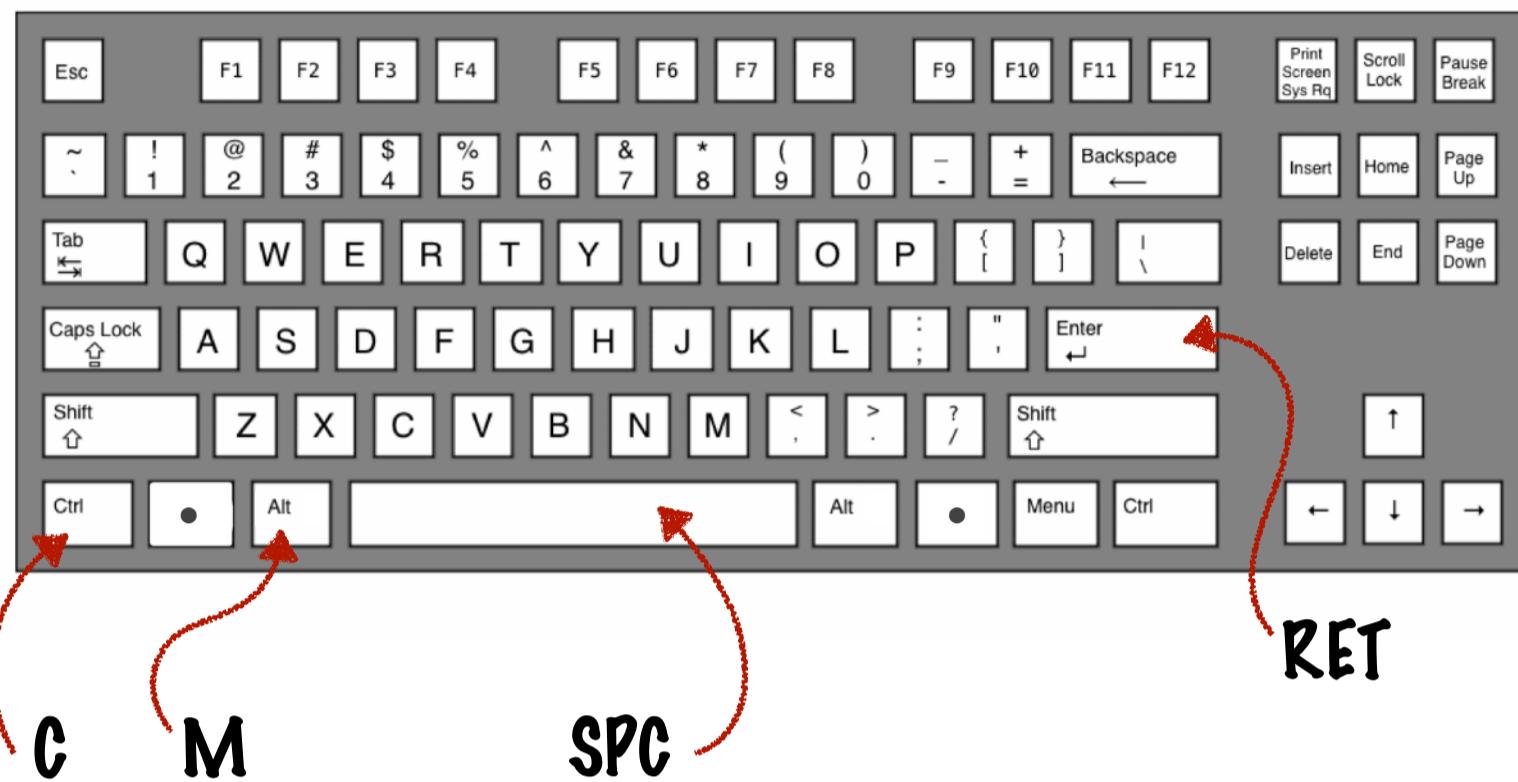
Emacs has a very long history and its own, formed over decades, terms, ways of solving problems, etc.

One of the key features of Emacs is its “keyboard-centric” design. Virtually any action can be executed without lifting your fingers from the keyboard. It may initially slow you down, but once you become accustomed to it, every action you take will be two to three times faster. This approach also aids in avoiding distractions caused by routine operations as they become ingrained in your muscle memory.

Of course, this principle isn't exclusive to Emacs. Even in Microsoft Word, it's more efficient to copy text with Ctrl-c and paste it with Ctrl-v than to navigate with the cursor to the corresponding icons.

In addition to the general Emacs hotkeys, there are specific commands for different packages. In the upcoming sections of the manual, we will explore

the commands utilized in howm<sup>4</sup>. The purpose of this chapter is to demonstrate how to decipher Emacs command notation.



The command `C-x C-s` denotes the following sequence of actions:

1. Press and hold the Control key, then press the `x` key.
2. Press and hold the Control key, then press the `s` key.

This command can be simplified, as both parts of it start with the Control key. Just press and hold the Control key, then press the `x` key followed by the `s` key.

Commands are case-sensitive. If a keyboard shortcut specifies a capital letter, that is the letter you should use.

Consider, for example, the command `C-u C-c , K`

This means you should press and hold the Control key, then press the `u` key followed by the `c` key. After that, type the comma and the capital `K`<sup>5</sup>.

The command `M-x` means: Press and hold the Alt key, then press the `x` key.

Why M? The letter "M" is short for Meta. There used to be a Meta key on some keyboards. Today, the Alt key is most often used instead, but the tradition of labeling it with the letter "M" has survived<sup>6</sup>.

<sup>4</sup> The list of these commands can be found on pp. 75-76

<sup>5</sup> Don't be afraid; most commands in Emacs are 2-4 characters long.

<sup>6</sup> The emacs-mac distribution uses the `⌘` key for Meta instead of the Alt key.

This command, by the way, invokes a special dialog—the minibuffer at the bottom of the screen.

Strictly speaking, any keyboard shortcut is a way to avoid typing lengthy commands. For instance, the keyboard shortcut `C-c , c` is responsible for creating a new note in howm. In this case, Emacs actually executes the “howm-create” command. If you invoke the minibuffer by `M-x` and type `howm-create`, it will achieve the same result.

In most cases, shortcuts do not need to be intentionally memorized. Through regular use, they tend to “stick” to your fingers naturally. This is especially true for common, everyday commands. Muscle memory develops quite quickly. It’s akin to learning fingering on the piano.

But of course, Emacs offers both icons and menus for those not ready to change habits. No pressure!

# INSTALLING HOWM

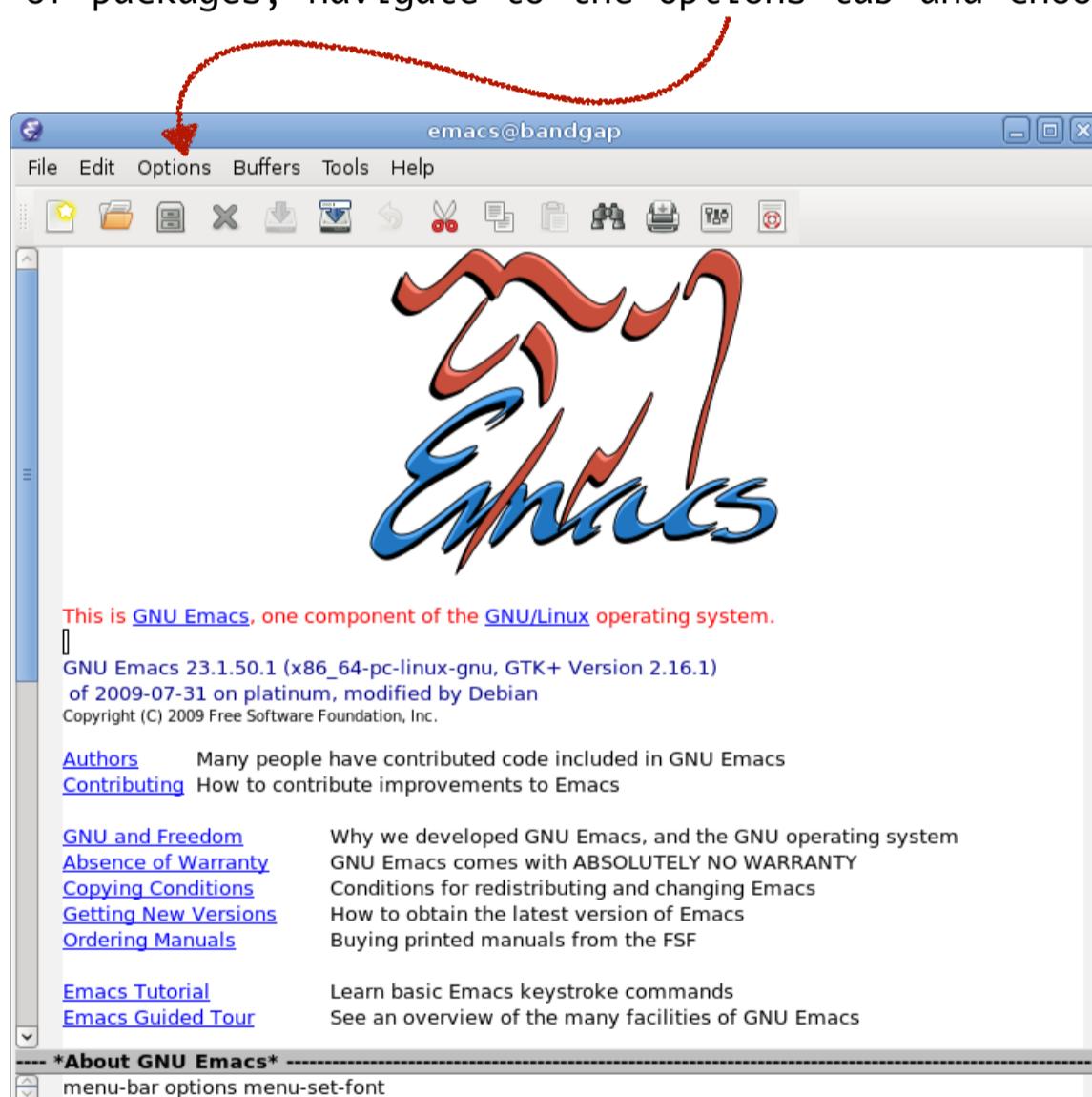
Even without any enhancements, Emacs stands out as an excellent text editor. Numerous intricate features are available immediately after installation.

However, Emacs features can be adapted to your needs thanks to additional packages.

Howm can be installed on Emacs just like many other extensions. So let's talk about the general techniques.

Before proceeding, let's first check which packages are already installed.

When you start Emacs, it opens a welcome screen, with a logo and introductory information. On the top, there is a menu bar containing various choices. To view a list of packages, navigate to the Options tab and choose “Manage Emacs Packages.”



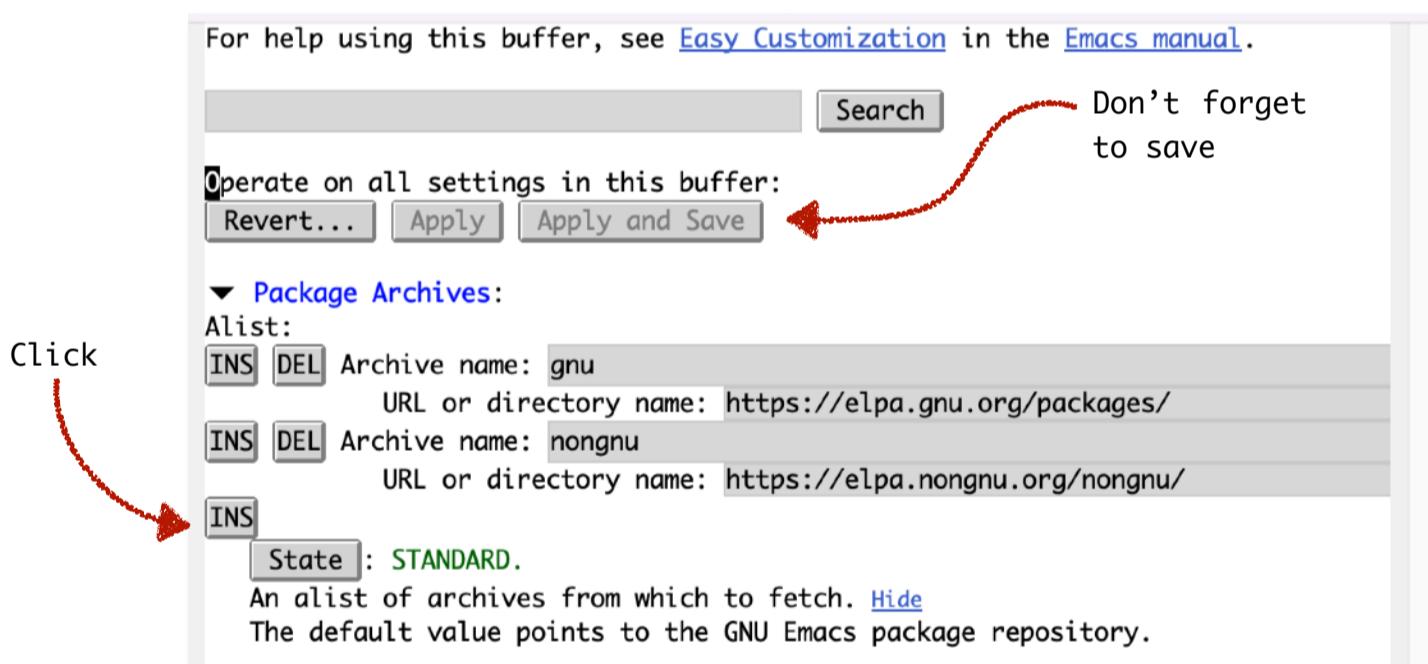
After pressing Enter on “Manage Emacs Packages,” a new window will appear, displaying an extensive list. The built-in packages will be at the very end of it. Feel free to install additional packages according to your needs, but do so thoughtfully to avoid cluttering Emacs with unnecessary features.

The ELPA repository (Emacs Lisp Package Archive) is accessible by default, offering numerous useful packages. However, some developers prefer the more

democratic MELPA (Milkypostman's Emacs Lisp Package Archive). We need to combine both lists within our Emacs setup since the `howm` package is hosted on MELPA.

Fortunately, there is a simple way to accomplish this.

1. Press `M-x`
2. Type `customize-variable` in the dialog box that opens below and press RET.
3. Type `package-archives` and press RET. You will be directed to the Package Archive page.
4. Click on `INS` button.
5. In the open form add Archive name **Melpa** and directory name:  
<https://melpa.org/packages/>
6. On the top of the page, press **Apply and Save** button.



Now that you have connected the MELPA archive with additional packages to your Emacs, examine the expanded list using the command: `M-x list-packages`.

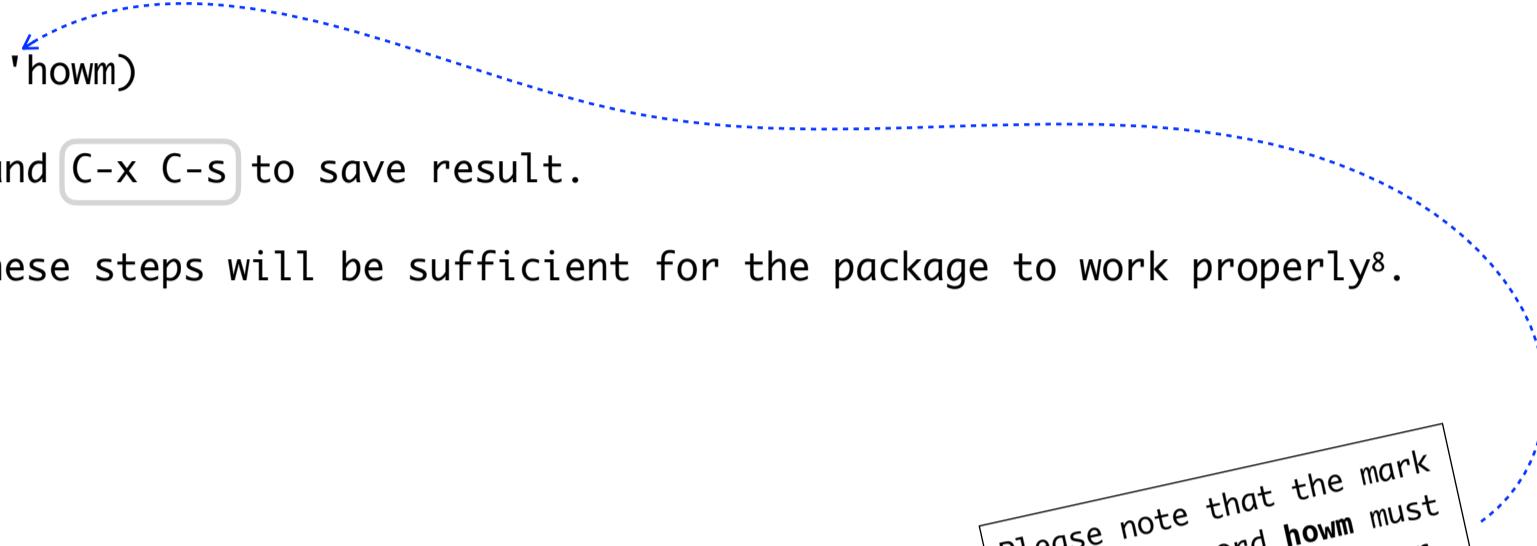
Installing and removing packages in Emacs is straightforward:

1. Press `M-x`
2. Type `package-install` and press RET
3. Type the name of the package - `howm` in our case - and press RET.

Now that the `howm` package is installed, we need to inform Emacs by configuring its activation.

1. Use command `C-x C-f`
2. Type `.emacs` and press RET
3. In open document<sup>7</sup> add the following line below others:  
`(require 'howm)`
4. Use command `C-x C-s` to save result.

Following these steps will be sufficient for the package to work properly<sup>8</sup>.



Please note that the mark before the word `howm` must be a straight typewriter apostrophe. A punctuation apostrophe ‘ won’t do.

---

<sup>7</sup> The `.emacs` file stores all the system modifications you have made in Emacs. If this file is deleted, Emacs will revert to its default settings.

<sup>8</sup> Some modifications may, however, improve package performance (refer to the “Advanced Settings” section).

# HOWM MAIN MENU

You can become familiar with the basic functions of howm in a matter of hours. If you need a quick crash course, refer to the official documentation on Github, where Kazuyuki Hiraoka has created a short guide.

You can start there, and then return to take a closer look at the nuances of using the system.

The first and most important thing to remember is the launch command for howm:

C-c , ,

This takes you to the main menu, which functions much like the captain's bridge. From here, you can easily navigate, issue commands, and monitor various compartments of your "ship."<sup>9</sup>

All the information doesn't fit in the window. You can swipe down by pressing the space bar or swipe up by pressing the backspace key<sup>10</sup>.

The underlined items are active. They serve as "buttons." Placing the cursor on them and pressing RET will execute a command or follow a link.

You can jump between "buttons" using the Tab key to jump forward and Tab with Alt to jump backward.

Basic cursor movement in Emacs is performed with the keys C-n - next line and C-p - previous line, C-f - forward, and C-b - backward.

You can, of course, move the cursor with the mouse or touchpad if you are more accustomed to it at this stage. However, the howm menu provides a much simpler and faster method for performing actions.

Each command has an assigned key (see illustration on the next page).

Pressing s will initiate a search, pressing c will create a note, etc.

To exit the howm menu, simply press the q key.

In this case, you will be redirected to the last non-howm buffer<sup>11</sup>.

---

<sup>9</sup> Of course, you don't have to go to the menu every time to interact with howm. It is more of a help system. Any actions can be performed using hotkeys as soon as you start Emacs.

<sup>10</sup> Standard Emacs commands C-v and M-v perform the same action.

<sup>11</sup> Simply put, Emacs will open the last non-howm file.

An expanded view of the home page of howm would look like this:

The diagram illustrates the expanded view of the howm home page with five callout boxes pointing to specific sections of the Emacs buffer:

- Menu Section:** Points to the top section of the buffer, listing basic commands like Remember, New, Dup, Search, etc.
- Tasks and Deadlines:** Points to the "Recent" section, which displays recently created or edited notes.
- Recent Notes:** Points to the same "Recent" section as the second annotation.
- Random Notes:** Points to the "Random" section, which lists random notes.
- Syntax Memos:** Points to the "Format of schedule and todo" section, which provides syntax rules for creating to-do lists.

**Annotations:**

- The menu section contains basic commands, which can also be accessed via hotkeys.
- This section includes a tasks, and deadlines etc. Howm incorporates a rather flexible system of organization and planning.
- Notes that you have recently created or edited are displayed here.
- Here will be a list of random notes. A chance to stumble across something you've forgotten.
- Memos about syntax for creating a to-do list.
- Here, you have the ability to customise the menu.

**Emacs Buffer Content:**

```
Emacs  File  Edit  Options  Buffers  Tools  Help
*howmM:%menu%*
<<< %menu%
e[Remember] c[New] D[Dup] Search(s[String] g[Regexp] o[Occur])
a[All] l[Recent] A[Around] y[Schedule] t[Todo] b[Buffers] x[Marks]
K[<Title] <Name] d[Date] i[Key] r[Update] w[Random Walk] [Killall]
.[Today] :[Yesterday] h[History] R[Update Menu] [Edit Menu] [Preference]

[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)
!deadline, +todo, -reminder, ~defer (top 50 entries)
-----
-----
-----

Recent
> 0000-00-00-00000 | = <<< %menu%
> 0000-00-00-00000 | = <<< %Editing Menu%

-----
-----
-----

Random -- If you do not like this, [Edit Menu] to erase it.
> 0000-00-00-00000 | = <<< %menu%

-----
-----
-----

Format of schedule and todo (Please replace {} with []):
{2002-10-21}@1 schedule -- (shown in schedule part. @3 = "3 days schedule to
10-23")
{2002-10-21}+7 todo -- float up slowly from the date in 7 days
{2002-10-21}!7 deadline -- float up fast from 7 days before the date
{2002-10-21}-1 reminder -- float at the date and sink slowly one unit per 1
day
{2002-10-21}~30 defer -- float at the date and repeat sinking and floating
with 30 days period
{2002-10-21}. done -- sink forever
(Numbers after marks are default values.)

-----
-----
-----

How to remember:
* Schedule at(@) the date
* Reminder sinks down(-).
* Todo floats up(+).
* Deadline needs attention!
* Defer waves(~) up and down.
* Done is the end(..).

-----
-----
-----

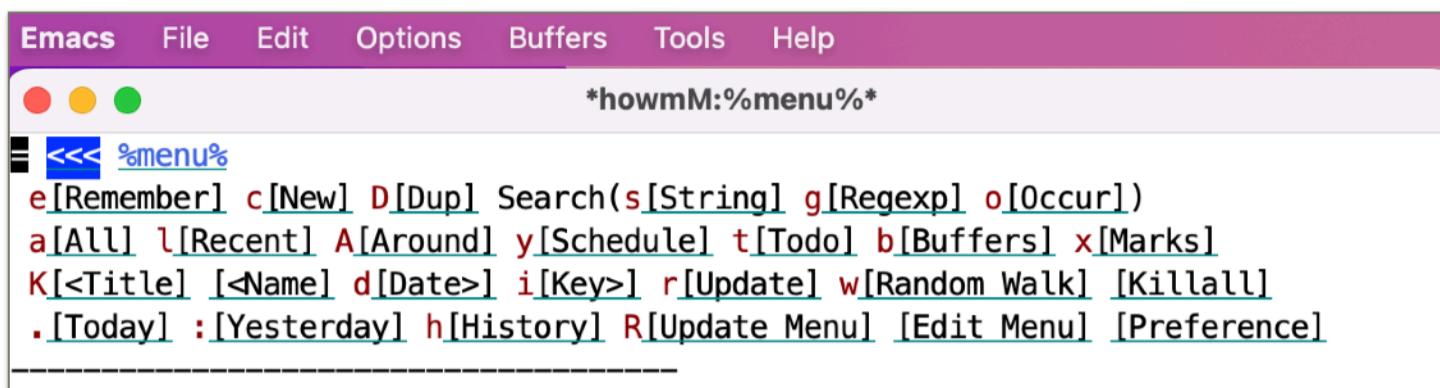
You can edit this menu itself.
>>> %Editing Menu%

-----
-----
[Update Menu] [Edit Menu]

-----
```

-:%%- \*howmM:%menu%\* Bot L24 (HM howm AL)

Now let's take a closer look at what actions can be performed from the main menu.



We'll go letter-by-letter, since each command is tied to a specific key. For now I'll give only a brief description of the commands, as we'll look at them in detail in later chapters.

- e** – creating a quick memo entry (opens a document without a template).
- c** – creating a new note.
- D** – note duplication. Handy if you use a complex templates. By default, the last entry in the list of open buffers is duplicated.
- s** – keyword search, plus the usual searches.
- g** – the usual searches, plus a regular expression search.
- o** – search in the last open buffer. That is, instead of searching the entire knowledge base, the search will be performed in a single note.
- a** – call the list of all notes (by default in the order of edits).
- l** – call up a list of notes you have recently worked with.
- A** – call notes in the order in which they were created.
- y** – calling up a schedule if you have something scheduled.
- t** – calling a to-do list.
- b** – call the list of open buffers.
- x** – open the last file in which you placed marks, and all marks will be listed in the upper window.
- K** – copy the title of the last note from the list of open buffers.

[<Name] – copy the numeric name of the last note from the list of open buffers, including the full path to the note.

**d** – insert the date into the note, specifically into the last entry from the list of open buffers. This function is particularly useful for planning.

**i** – paste keywords into the last entry from the list of open buffers.

**r** – update notes that have “come-from” links. If keywords appear in some text, they will be activated<sup>12</sup>.

**w** – view notes by following links randomly. The cycle is endless. You can stop jumping by pressing **C-g**.

[Killall] – removes all howm notes, except unsaved ones, from the list of opened buffers.

. – opens a list of notes that have been created today.

: – opens a list of notes that were created yesterday.

**h** – call a list with the history of search queries.

**R** – update menu (refilling sections with todos, recent, and random files). In fact, any call to the menu updates it. This command is useful if you have the menu open in a separate frame or tab.

[Edit menu] – go to the menu settings.

[Preference] – go to the howm settings.

As I said, all commands located in the menu are also available thanks to hotkeys. They can be invoked even while working with other documents outside of howm<sup>13</sup>. Just trigger them by using the keyboard shortcut **C-c**, followed by the character specified in the menu.

For instance, the “Random walk” function is invoked with the command **C-c , w**.

---

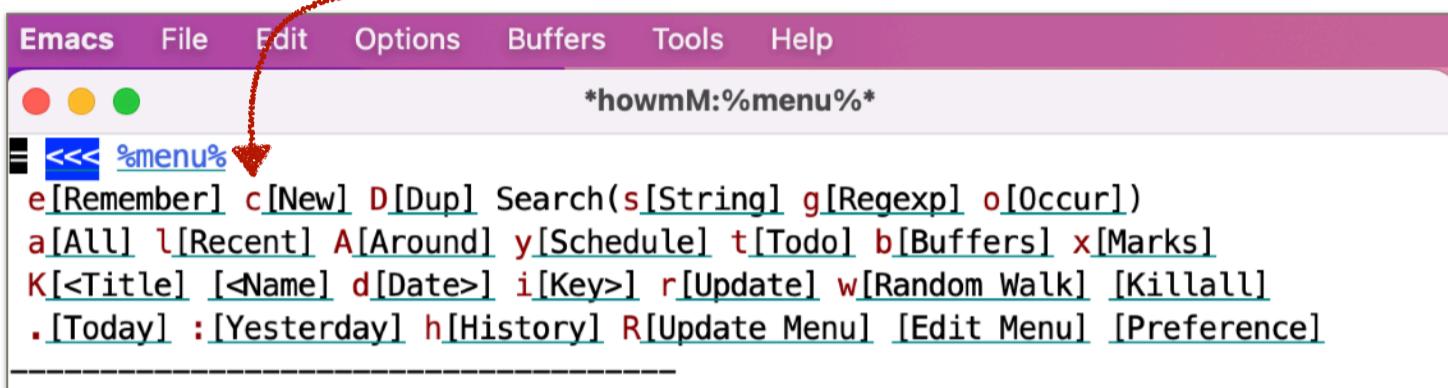
<sup>12</sup> More on this in the chapter “Keywords in howm.”

<sup>13</sup> Howm responds to commands from major modes. If you are running some minor mode, you need to activate howm (**M-x howm-mode**). In general, if a command doesn’t work, check if howm is activated.

# CREATING NOTES

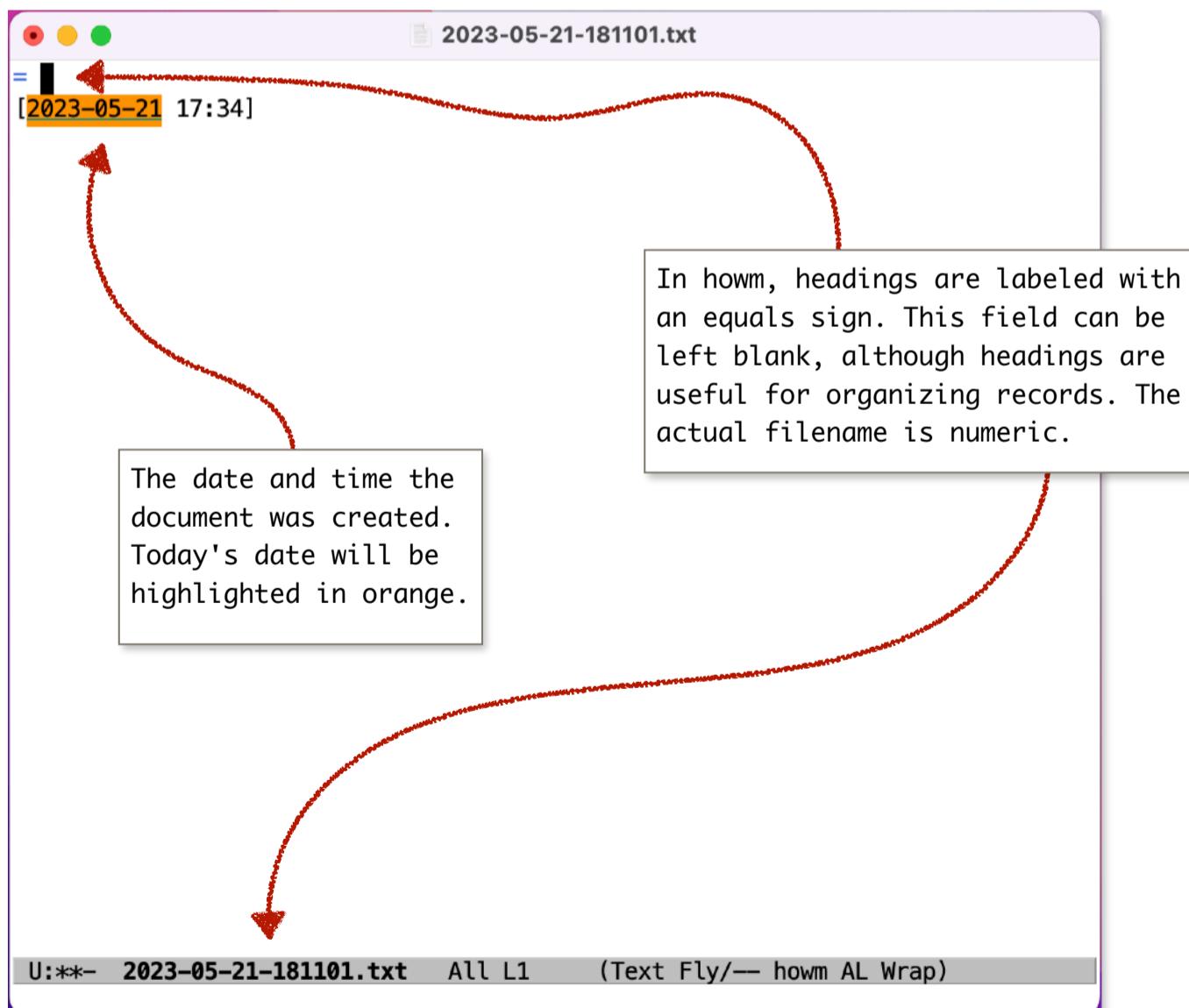
Let's create a first note in howm.

If you are in the main menu, it is enough to move the cursor to the corresponding [New] button or just press the **c** key. It naturally corresponds to the initial letter of the word “create.”



You can perform the same action, bypassing the main menu, with the **C-c , c** command. After starting Emacs, type this keyboard shortcut, and howm will open a new note for you.

What do we have here?



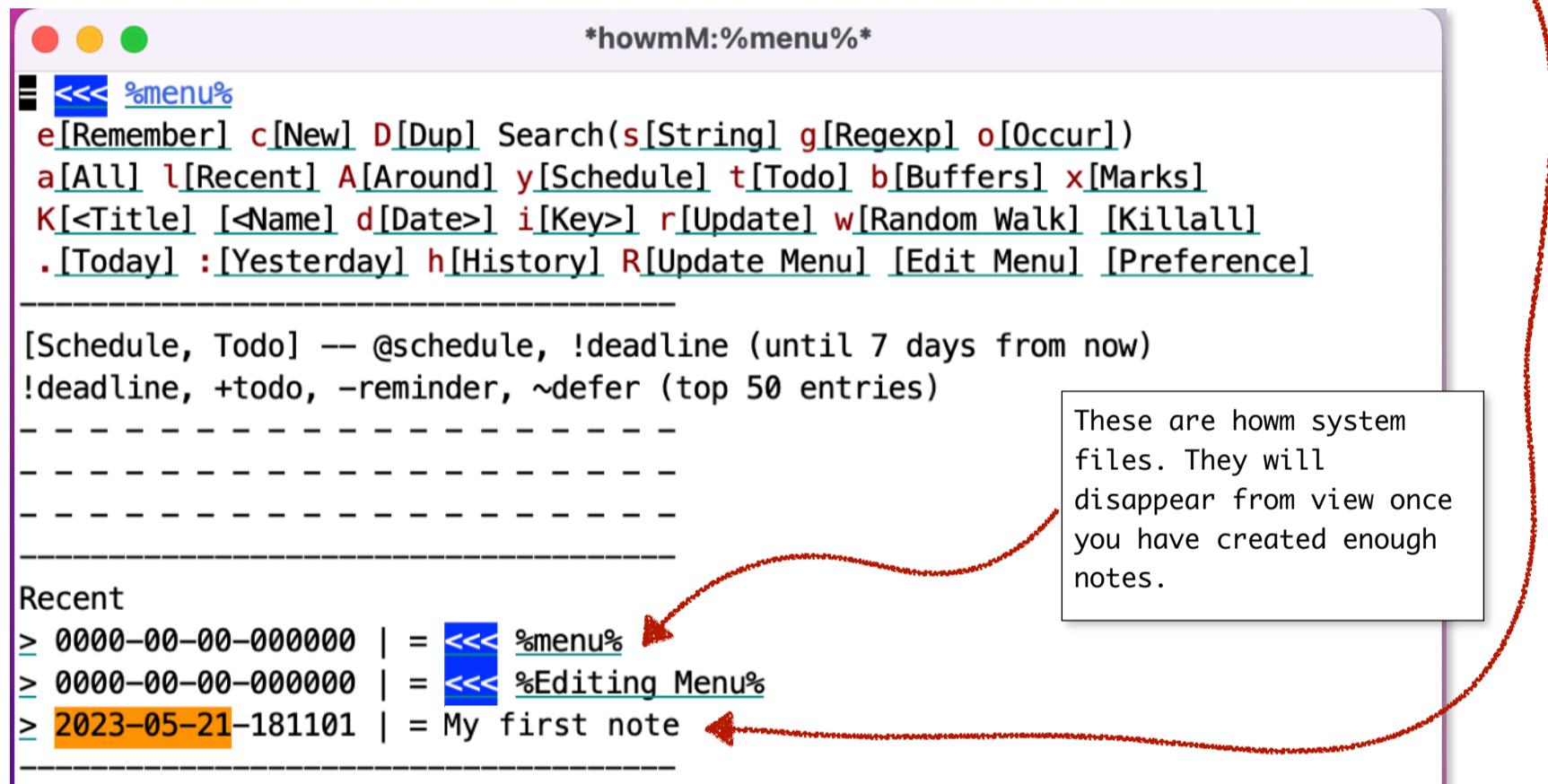
The unique identifier of a note is the numeric name. Therefore, the heading can be the same for two different notes (although this is bad practice) or none at all.

Move your cursor below the date and start typing. It's simple as that.

By default howm saves records in txt format, so no “cosmetic manipulations” with text (making bold, underline, etc.) are possible<sup>14</sup>.

After you have written your first note, save it with the standard Emacs command - `C-x C-s`

Checking the main menu, you'll notice that it appears in the Recent list.



To return to a note from the main menu, move the cursor to the corresponding line in the “Recent” section and press RET.

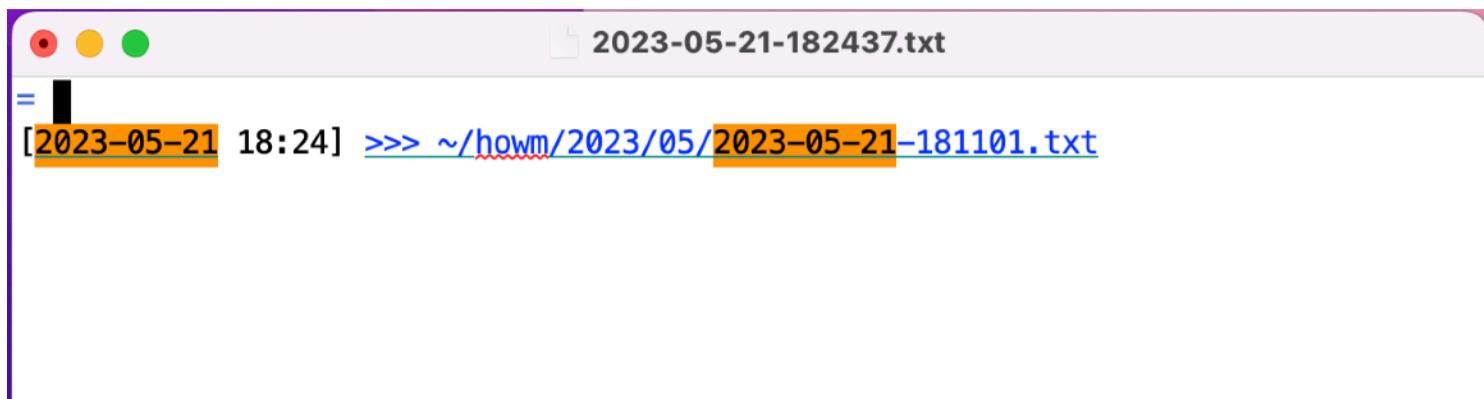
As you write your first note, you may be visited by ideas that further develop the topic.

It is tempting to write them immediately, turning the text into an endless scroll. It is better, however, to keep notes modular.

While you are still in the first note, create another one using the same command - `C-c , c`

<sup>14</sup> You can change the file extension (see “Advanced Settings” p. 68). Additionally, if you place an org document in the home folder, its specific formatting will still function. Howm is friendly with any minor modes.

Notice that the new note template is slightly different. A link to the previous note appears next to the date<sup>15</sup>.

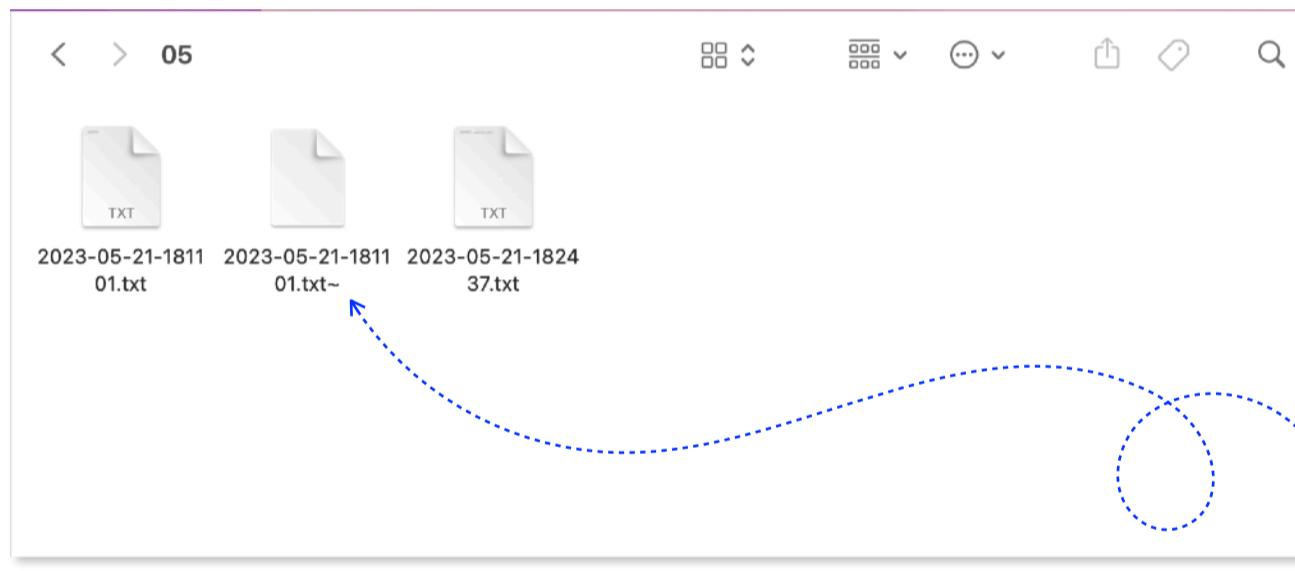


Whenever you, while in one note, create another note, howm makes a link between them<sup>16</sup>. A handy feature for associative thinking.

If you don't need this link, use the undo command - `C-/`

If you change your mind about creating a note at all, use undo again.

Speaking of saved notes, they are stored in the `howm` folder in the home directory. Howm creates a subfolder for each year and a sub-subfolder for each month<sup>17</sup>.



Next to each file, you may notice its clone with a strange tail. Don't be alarmed - it is a normal Emacs practice to create a backup copy. This practice is very useful if something goes wrong. These files will be hidden and not bother you. We looked into this folder only for excursion purposes.

At some point, you will have a lot of notes. How do you find the right one? How do you make an overview of your notes? Howm manages these tasks with the help of lists.

<sup>15</sup> How links work in howm will be discussed a little further on.

<sup>16</sup> The link only appears in the second note.

<sup>17</sup> The full file path looks like this: `~/howm/year/month/file name`. However, this is all behind the curtain of the program, and the folders are not visible in the howm interface.

# WORKING WITH LISTS

The commands for calling recent notes, calling all notes, searching, and even navigating through links switch howm into a dual-window screen. This interface is central to the majority of interactions with notes. Let's break down its "anatomy."

The screenshot shows the Howm application interface. The top window displays a list of notes with their titles. The bottom window shows a preview of the note currently selected in the list.

**Top Window (List View):**

- 2023-04-14-190528.txt | = Philosophy as Psychotherapy
- 2023-04-14-182646.txt | = The Allure of Self-Improvement
- 2023-04-14-184903.txt | = The Diary and the Challenge of Self-Discovery
- 2023-04-14-183747.txt | = Human Existence
- 2023-04-14-183154.txt | = The Imperfection of the World
- 2023-04-14-182224.txt | = No one is worse than
- 2023-04-14-182039.txt | = Avoid Torturing Yourself
- 2023-04-14-160713.txt | = The Ten Commandments of Language
- 2023-04-14-133721.txt | = Another Language, Another Persona
- 2022-10-06-111115.txt | = Accumulation of Knowledge
- 2023-04-13-144303.txt | = Realistic Deadlines in Language A
- 2023-04-13-142613.txt | = Minimum Requirements in Language
- 2023-04-13-143504.txt | = Thoughts on Language Learning
- 2023-04-13-143407.txt | = Vocabulary as a Potential Obstacle
- 2022-10-18-143554.txt | = Computer Languages
- 2023-02-13-120000.txt | = Language Learning and Academia
- 2023-04-13-142326.txt | = Reading as the Primary Tool in Language Learning

U:%%- \*howmS\* 25% L192 (HowmS)

**Bottom Window (Preview View):**

No one is worse than  
[2023-04-14 18:22] >> ~/howm/2023/04/2023-04-14-182039.txt

If one is mean to himself, to whom will he be generous?  
He will not enjoy his own riches. No one is worse than one who is grudging to himself... (Ecclesiasticus 14:5-6)

**Annotations:**

- A red arrow points from the left side of the top window to the column of note names.
- A red arrow points from the right side of the top window to the note titles.
- A red arrow points from the bottom window to the preview text.

**Bottom Window Footer:**

U:%%- 2023-04-14-182224.txt All L1 (Text Fly/-- howm AL Wrap)

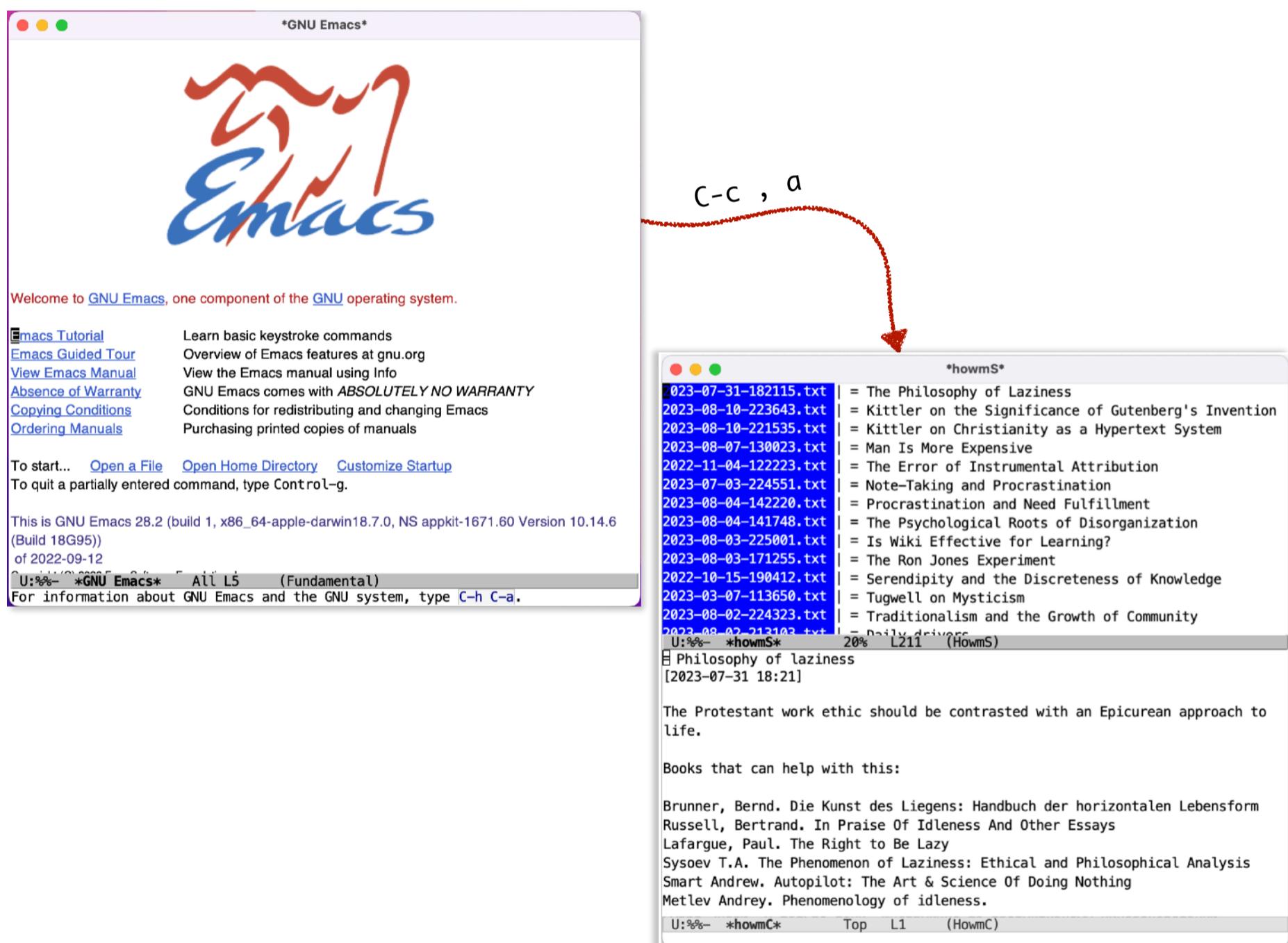
Since lists are a key element in working with howm, it's worth taking a closer look at the features and functions hidden here. Regardless of which list you have in front of you, the principles of working with it will be the same.

From the main howm menu, the note lists can be accessed using the commands:

**a[All]** **l[Recent]** **A[Around]**

However, you don't have to visit the main menu to access any of these lists. It is much more convenient to use special commands for this purpose.

Let's open a list of all the notes - **C-c , a**



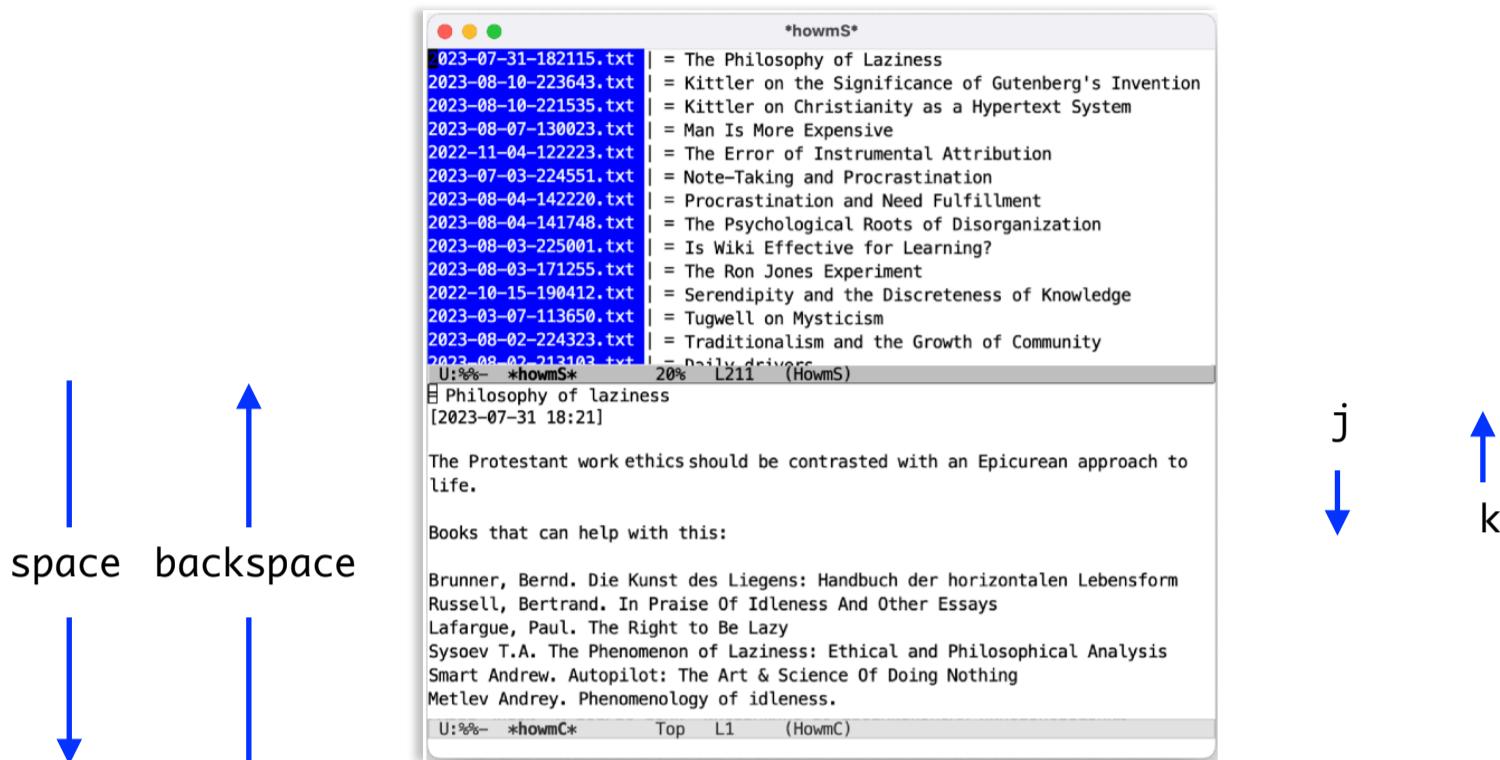
Let's talk about how to move around.

Move the cursor through the list using the **n** and **p** keys, down and up, respectively<sup>18</sup>. The bottom window flips through the notes as you move the

<sup>18</sup> The standard Emacs commands **C-n** and **C-p** work too. It's best to use them for practicing muscle memory.

cursor around the list. You can also use the Tab key to move to the next line in the list and Tab with Alt to move up.

Often, the entire content of a note does not fit in the preview window. You can scroll down by pressing the spacebar and up by pressing Backspace<sup>19</sup>. Use the j and k keys to scroll slowly through the text line by line.



If you want to jump to the bottom window, press RET. The cursor will then move to the note, allowing you to make any necessary additions or edits.

Notice that the top window with the list will remain open. This is useful for all sorts of quick manipulations with text. You can return to the upper list using the C-x o command, which is the conventional Emacs way of switching between windows.

Sometimes, you may want to remove the upper list and focus solely on the note content. Move the cursor to the line with the target note and press C-u RET<sup>20</sup>.

If you want the exact opposite – no preview window and a clear overview of the list of notes – you can close the bottom window by simply pressing 1. To restore the preview, press 2 on your keyboard<sup>21</sup>. Alternatively, you can switch back and forth by pressing v.

Now, let's turn directly to working with lists.

<sup>19</sup> On a Macbook, it's the delete key. Do not be alarmed in this case, it will not delete anything.

<sup>20</sup> You can also jump to a note with RET and then press C-x 1. This is a standard Emacs command that closes all windows except the current one.

<sup>21</sup> To get more from this function, you need to switch off notes preview while scrolling through the only list screen. More details on that can be found in “Advanced Settings” on page 69.

As mentioned earlier, the command `C-c , a` brings up a list of all the notes stored in howm. By default, they are arranged based on the order of last edit. This means that the files you have recently worked with will be at the top<sup>22</sup>.

With an entire list of notes, you can perform several useful operations.

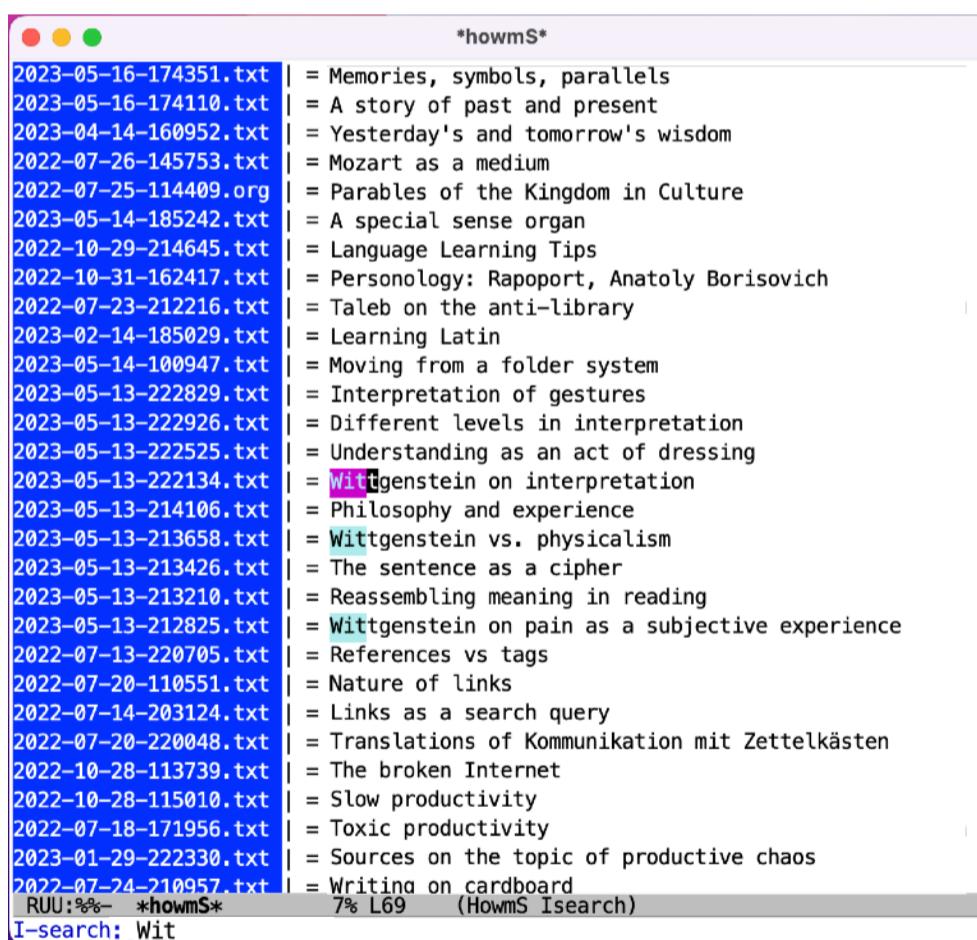
### Counting notes

Type the line count command - `C-x l` to find out how many notes are in your knowledge base. The number will appear in the minibuffer<sup>23</sup>.

### Quickly navigate to notes

The entire list can be useful for quickly jumping to a note, especially when you remember part of its title. In such cases, it's more convenient to switch to the “list only” view.

Press `C-s`, the standard Emacs text search, and start typing the name of the note. You don't even have to start with the first word. Use the same `C-s` command to jump between search results.



In addition to the entire list, you can request a list of recent notes - `l [Recent]` in the main menu.

<sup>22</sup> The likelihood that you will need those files again is very high. This is the core principle of Yukio Noguchi's system.

<sup>23</sup> Since there are two lines from the system file in the entire list, subtract them from the resulting number.

If you are not in the main menu, use the keyboard shortcut - C-c , l

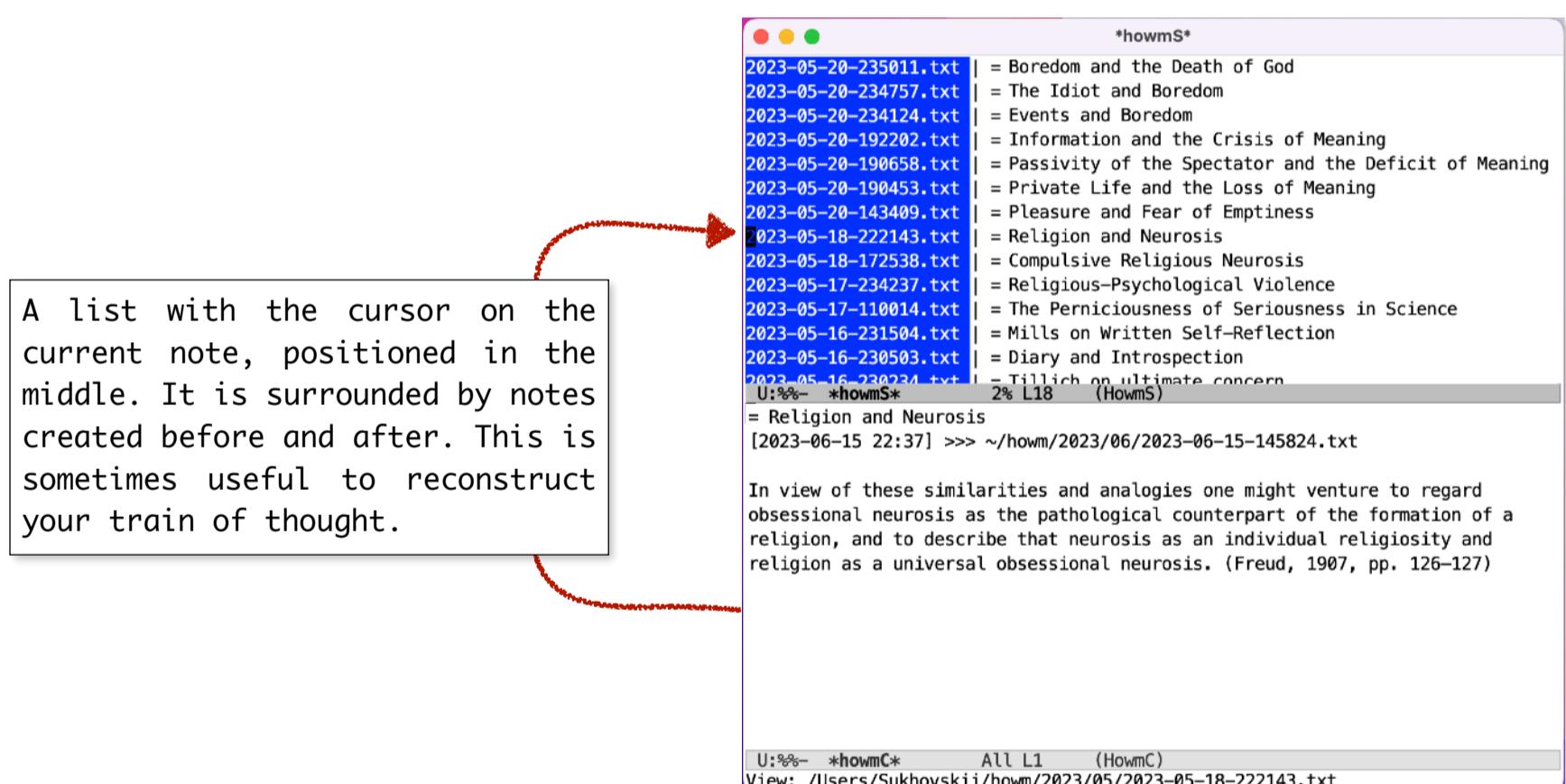
With this command, howm will display notes created or edited in the last seven days<sup>24</sup>. This list allows you to quickly switch between the notes you are currently working with.

Do you need a longer list? Here is an example of how you can request a 20-day summary from howm: C-u 20 C-c , l

By combining variations of this command, you can view the files you worked with this month or quarter. This is useful for summarizing results and self-analysis.

The command A[Around] in the main menu brings up a list of notes sorted by date of creation.

While in an open note, you can query the same list with C-c , A. The difference is that the list will be contextual, centered on the current note.



What if you want to see all the notes entered into the database today? They are already highlighted in orange. However, if you want a separate list, use the C-c , . command.

Keep in mind, that howm will display only the list of created notes; edited notes are not counted in this case.

<sup>24</sup> If you make a request on a Thursday, the list will include all changed/created notes from the previous Thursday to the current Thursday.

If you use links to a numeric filename, the list will also include old notes linked to today's notes. It is because howm searches for notes by the date of creation in both the filename and the text itself.

Older notes will be highlighted in blue, while today's notes will have a white background.

The screenshot shows a terminal window titled '\*howmS\*' displaying a list of notes. At the top, there are two entries:

```
2023-03-24-223958.txt | = Who are you?
2023-05-23-205541.txt | = From dialog of subpersonalities to silence
```

The main area shows a note from March 24, 2023, at 22:39:

```
U:%%- *howmS*      All L1      (HowmS)
└ Who are you?
[2023-03-24 22:39]

The marvelous TV series "Dirk Gently's Holistic Detective Agency" – which has
almost nothing to do with the book – has some entertaining dialog.
It's the second season, episode "Shapes and Colors".
A caterpillar from a parallel world (Alice in Wonderland reference?) asks
Amanda:
- Who are you?
- Amanda Brotzman?
- That is just a noise people can make to refer to you. You're not your name.
```

At the bottom, there is a status bar:

```
U:%%- *howmC*      Top L1      (HowmC)
View: /Users/Sukhovskii/howm/2023/03/2023-03-24-223958.txt
```

To see a list of notes created yesterday, type the command `C-c , :`

You may be interested in notes that you created some number of days ago (let's take 50 days as an example).

Then the command `C-u 50 C-c , .` comes to the rescue.

As you can see, interacting with lists is central to working in howm.

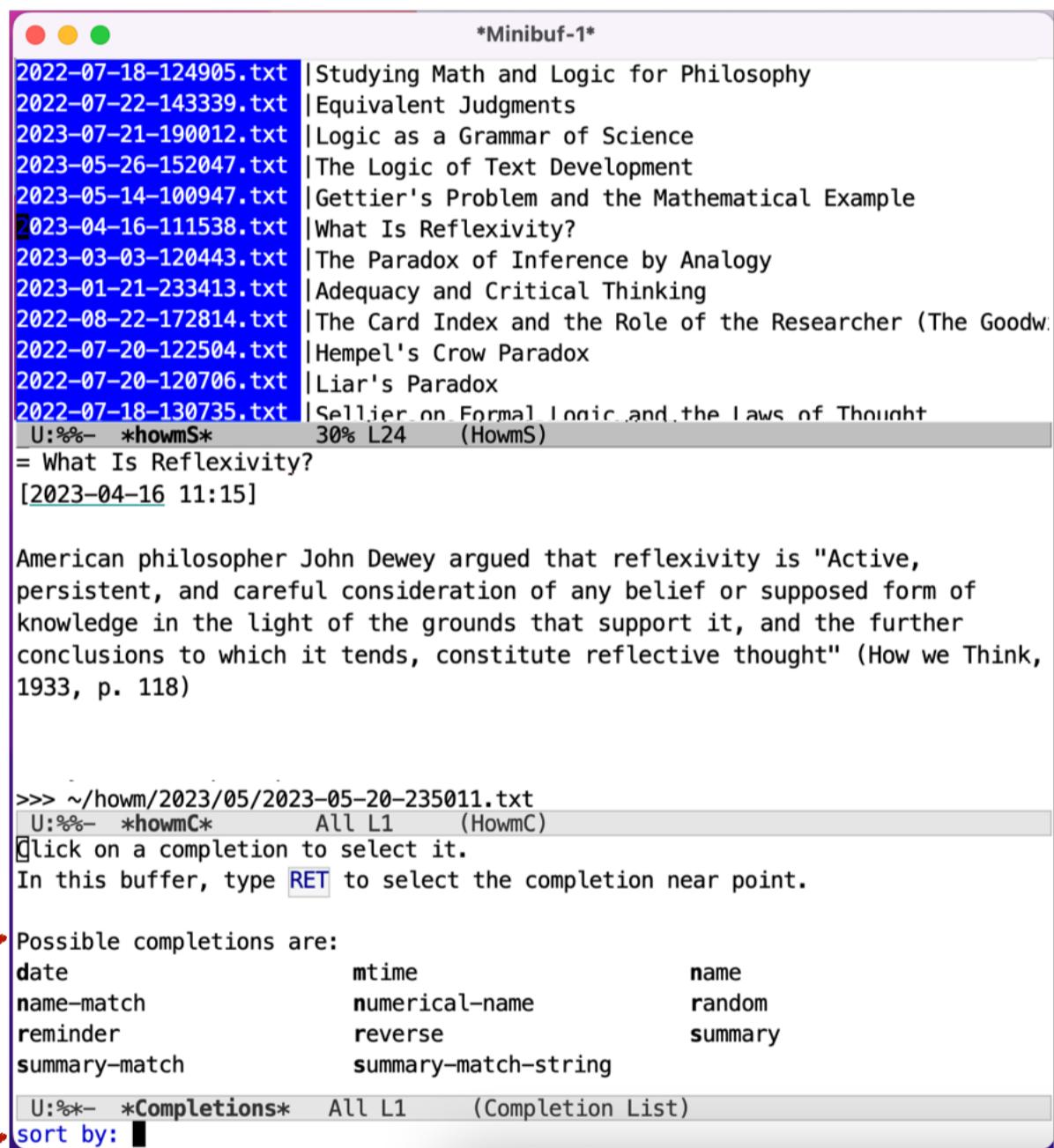
# SORTING NOTES IN LISTS

When you ask howm for a list of notes with the command `C-c , a` or `C-c , l`, you get it sorted in order of recent edits. That is, the notes you have worked on recently will be at the top.

If you want different ways of representing information, press the S key.

Howm will prompt you to choose the criteria for listing your notes.

Next, press Tab to see the options<sup>25</sup>.



You may not need all the options, but some can certainly make navigating through your notes more efficient.

## date

The function allows you to create a list of notes sorted by the date of creation, with the most recent notes at the top.

<sup>25</sup> The Vertico package can enhance your interaction with the minibuffer.

To see the list in reverse order (from oldest to newest), press **R**. To go back, press **R** again. The **R** command reverses any list, placing what was at the bottom at the top.

If you want an immediate reverse order of edits, creation dates, or alphabet, type **C-u S** and then select the desired option.

### **reverse**

Performs the same function of turning the list upside down. It is easier and faster to use the **R** key.

### **summary**

To the right of the numeric file name in the list is the heading (if you have created one). The “summary” command is needed to sort notes alphabetically by headings.

In the titles, as you might guess, you can add an additional layer of metadata. By using certain prefixes before the note’s name, you can create markers for note status or construct sequences of thoughts.

### **mtime**

If, after all these rotations, you want to return to the list based on the edit date, select the **mtime** (modification time) command. Alternatively, you can simply restart the list.

### **random**

Tired of order? Shuffle your notes using the **random** command. You might come across a forgotten but interesting note<sup>26</sup>.

### **name-match**

With this command, you can arrange your notes in the order you like<sup>27</sup>. Howm will prompt you to specify the name of the file to move to the top of the list. In this case, it is a numeric name. You can specify only part of it.

### **summary-match**

The same reordering operation can be performed based on headings. Note that you can specify part of a heading, such as a word, and howm will move up all the notes that mention that word in their headings.

---

<sup>26</sup> As a reminder, there is already a list of random notes on the initial howm page.

<sup>27</sup> The manual arrangement in the list is very useful for editing notes and stitching them together into a single text, which we will discuss further on.

## **summary-match-string**

For those who understand regular expressions, there is the possibility of more tricky filters<sup>28</sup>.

## **name**

This command allows you to sort files with names other than numeric. Where can this be useful?

Notes created in `howm` are automatically given a numeric name based on date and time. However, if you want to include files with descriptive names, `howm` will easily accept them. The “name” command alphabetizes such notes, placing them at the end of the list<sup>29</sup>.

## **reminder**

The command allows you to bring tasks, if any, to the top of the list. We will revisit this function when discussing organization in `howm`. The time management system here is quite powerful.

## **numerical-name**

A command that lets you sort your correspondence if you decide to integrate a knowledge base with an email client<sup>30</sup>.

---

<sup>28</sup> The usage of Regex is a special topic that is beyond the scope of this brief tutorial.

<sup>29</sup> You don’t even have to put these strangers in `howm` folder. You can simply specify the path to them. See “Advanced settings,” p. 69

<sup>30</sup> If you’re using something like Wanderlust, you can configure `howm` to search for files from the respective directory. Refer to “Advanced settings” for details.

# WORKING WITH LINKS

A knowledge system is more than just a collection of notes; it is a network of references between them. In howm, these connections are implemented through hyperlinks. However, the mechanism of their operation is not quite ordinary.

You don't instantly jump from one text to another. By moving the cursor to the link and pressing RET, you initiate a search. The screen divides into two parts: the upper part shows the search results, and the lower part previews the note on which the cursor is in the list. This intermediate stage allows you to examine other links to the target note.

Let's explore specific examples.

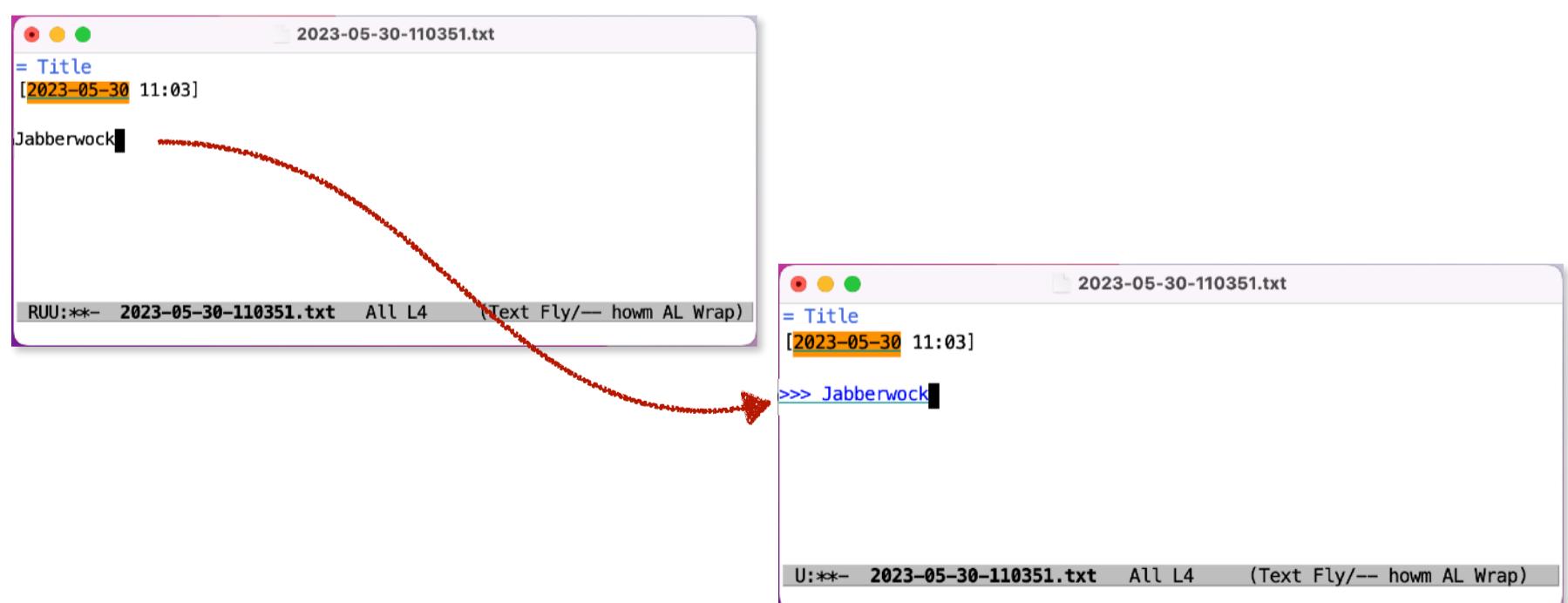
The syntax of links in howm is extremely simple. There are two special signs for links:

>>> Three right-pointing arrows<sup>31</sup>

<<< Three left-pointing arrows.

They produce different outcomes, so it's worth considering them separately. In this chapter, we will discuss the first type, and in the next chapter, we will cover the second type<sup>32</sup>.

Adding a “spruce-sign” in front of a word or phrase creates a link. Better add a space character after the symbol for better readability.



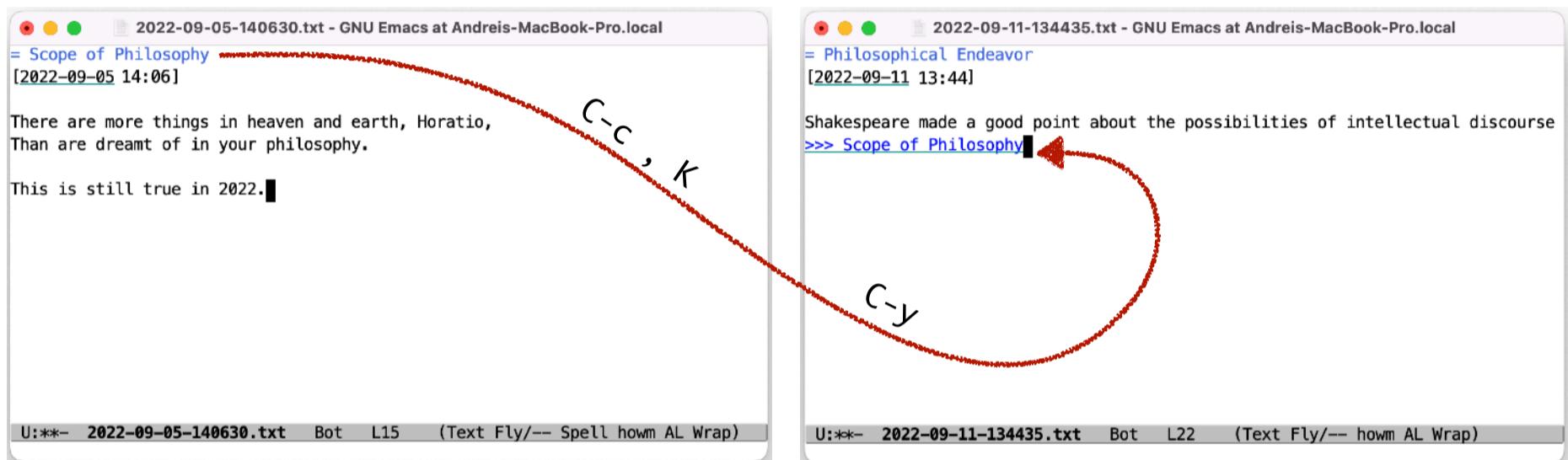
<sup>31</sup> If you like both emoticons and the metaphor of evergreen content, you can consider it as a symbol of a spruce.

<sup>32</sup> You can also use double square brackets, similar to wiki-style markup. More on this in the next chapter.

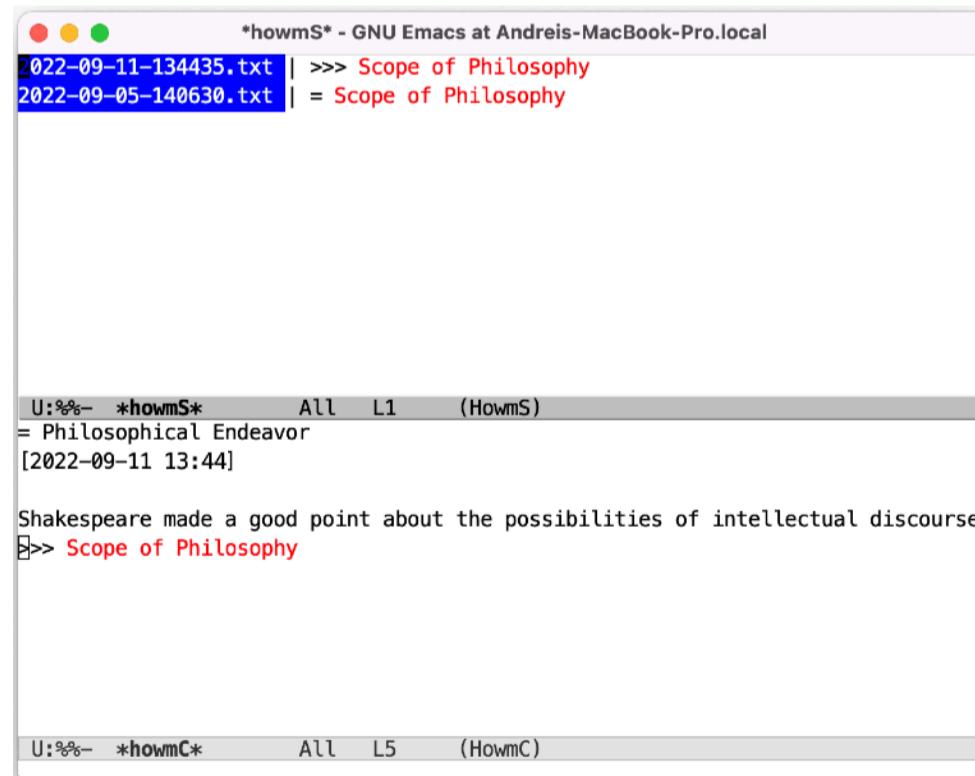
Anything on the line to the right of the `>>>` symbol in howm will be interpreted as a link/search trigger. Therefore, you need to start a new line to continue regular text after the link.

You can create a link to a specific heading or a numeric file name. The result, though, will be slightly different.

To create a link to a heading, simply write it after the `>>>` sign. It's easier to do this from the target note. Just press `C-c , K` to copy the heading, then go to another note, type `>>>` sign, and use the command `C-y` to paste the heading. Now, a link is created.



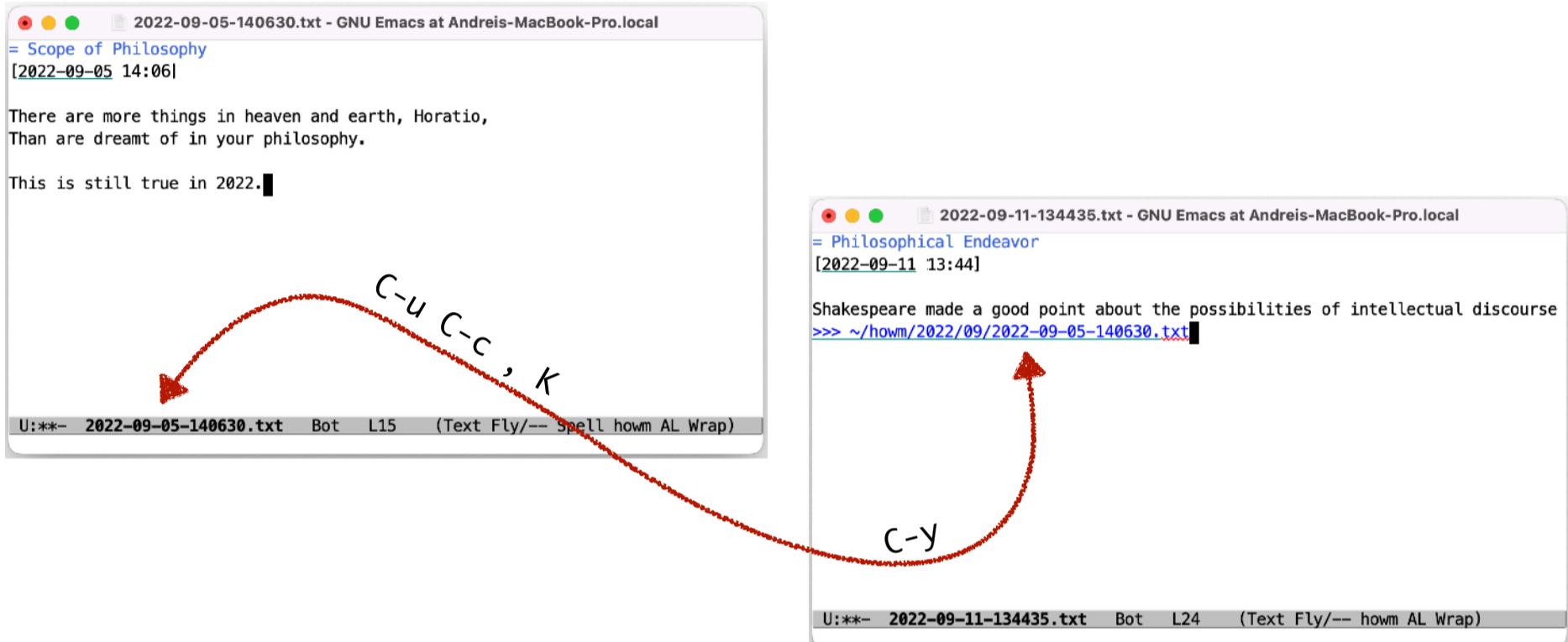
Let's try it out. Place the cursor on the link and press RET. You will be forwarded to the split-in-two screen (list and preview window). Here, you'll see a note with the link `>>>` and another note with a title (indicated by an equals sign).



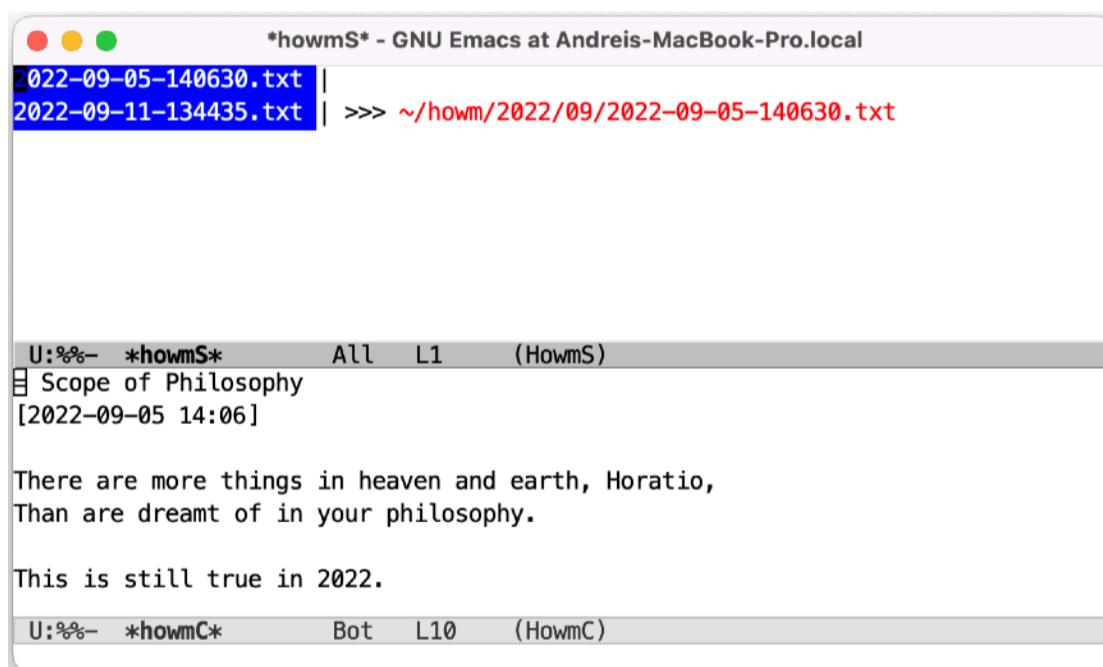
In fact, howm displays a list of files where the phrase "Scope of Philosophy" occurs. The listing includes all notes that contain this phrase anywhere, whether they are headings or just mentions.

Another way to create a link is to use a numeric file name. Such a link is more reliable. For example, in the previous case, if you change the title, you will encounter broken links. A numeric file name serves as its unique and unchangeable ID, independent of the content.

With the command `C-u C-c , K` howm will copy the absolute path to the file. Paste it into a favorite note, preface it with a `>>>` sign.



When you follow the link, you'll get a slightly different list of notes.



A note referenced by other entries will always be at the top<sup>33</sup>. This is handy because you can follow the link by simply pressing RET twice. Another argument in favour of such links is that they are the ones used in howm when you create a chain of notes (see p. 17). Of course, just the dates in the list are not very informative. To display the headings, press `T` and to switch back press it again.

<sup>33</sup> If you create links on headings, the list is formed differently, and the referenced note may end up in the middle or at the end.

In fact, the absolute path to the file lets you use the Emacs standard hyperlink system. You can instantly jump from one note to another<sup>34</sup>.

Instead of >>> sign, type this prefix **file://**

**file://~/howm/2023/09/2022-09-05-140630.txt**

It is better, I think, to become accustomed to the howm (link-list-transition) approach. It has a number of advantages when working with notes. First of all, it's an opportunity to remind yourself of other trajectories in the hyperlink system.

Also, consider when to use the numeric file name as a reference and when to use the heading. However, it is preferable to maintain consistency. When you click on a link with a numeric name, you will get a list of all notes that contain the same link. Nevertheless, howm will not include links to the heading of the same note, and vice versa.

The advantage of linking to the title of a note or any other word combination in the text is that you see unrecorded but potentially important relations. The downside is semantic noise if there are too many items in the search result.



By using a numeric file name, you can organize backlinks where they are needed. To do this, simply place a link to the note itself.

1. Copy the file name with the command **C-u C-c , K**
2. In the same note, type >>> and press **C-y**
3. Clicking on the link will open a list of notes that contain the numeric name of the current file (refer to it).

Howm recognizes usual links to internet pages. Any link copied from the browser address bar will work fine<sup>35</sup>.

If you need to link to a file outside the howm directory, use the previously mentioned prefix: **file://**

---

<sup>34</sup> However, with this transition howm will not recognize the note as native (unless it is already open in the buffer). This means that the howm syntax will not work either. You will have to activate it additionally: M-x howm-mode

<sup>35</sup> You can view internet pages inside Emacs, but this requires additional settings; otherwise, the link will open in your default browser.

Next, you need to specify the full path to the file. For example:

file:///My\_brilliant\_thoughts/Nothing\_yet.org

Any plain text files (such as txt, md, org) will open without problems inside Emacs. Pictures will also open. For more intricate formats like Docx, Pdf, DJVU, you may need additional plugins and Emacs settings.

The ability to weave a web of knowledge that connects documents in different directories is a huge advantage of Emacs. However, keep in mind that moving a file to a different folder or renaming it will break the link<sup>36</sup>.

What if you want to reference a howm memo from another Emacs document?

For example, I don't keep book summaries in the howm knowledge base. However, I often need to mention some ideas from my notes in the summary. A link is highly desirable.

I will show you how to create an external link to a howm note, using Org-mode as an example. The reason for this choice is simple: Org-mode is one of the best tools for working with long texts<sup>37</sup>.



1. Open the note you want to reference.
2. Open another Emacs frame - `C-x 5 2`
3. Open the desired Org file in a new frame.
4. Go back to the note - `C-x 5 o`
5. Place the cursor on an empty line and click `C-c l`
6. Go back to the Org document
7. Insert a link - `C-c C-l`

I wouldn't say that the linking system in howm is the most ergonomic, but it is quite functional. You can get used to it and adapt it to your tasks.

One limitation of howm when working with links is immediately apparent: there is no autofill for this task. You have to find both notes you want to connect

---

<sup>36</sup> It is best to develop for yourself a simple and consistent system of naming files and creating directories. There are no universal solutions, but generally, the less nested, the better.

<sup>37</sup> Refer to "Advanced Settings" to learn how to pair howm and Org-mode. See p. 71

and copy the address of one to the other. For people spoiled by automation, this may seem like a terribly slow process.

In fact, once you understand the algorithm, the process will not pose any difficulties. It is most convenient to leverage the potential of lists already available in howm.

Any note that has been modified is moved to the top of the list. Knowing this, you can easily place the required notes next to each other.



1. Open Note A, press space and then backspace, and save the “changes” - `C-x C-s`
2. Locate Note B using the command `C-c , s`
3. Copy either the numeric name of the file or the title
4. Bring up a list of recent edits - `C-c , l`
5. Note A will be at the top. Navigate to it and paste the link.

This life hack applies to old notes that you've decided to link. If you are already working with Note A, chances are high that the first point with fake changes is unnecessary.

As entries grow, there may be several links in one note, and in different paragraphs. You can move between them very quickly.

`C-c , n` – Next link.

`C-c , p` – Previous link.

In the next chapter, we'll talk about the function of <<< sign.

# KEYWORDS IN HOWM

What is the purpose of <<< sign?

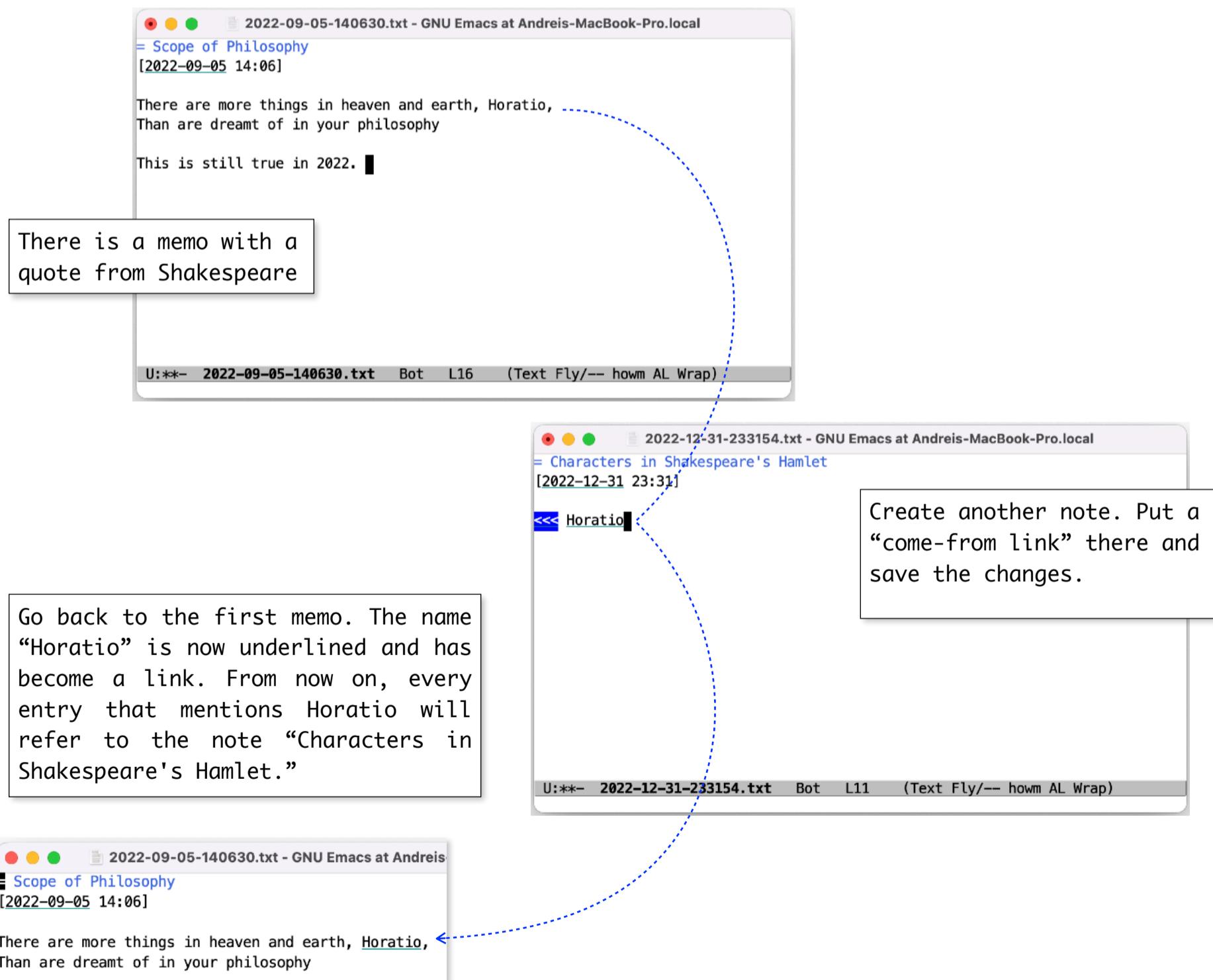
This is also a prefix for a link, but with different properties.

The spruce pointing to the right >>> symbolizes a transition from the current note to another note, i.e. the link leads from here. This is “goto link.”

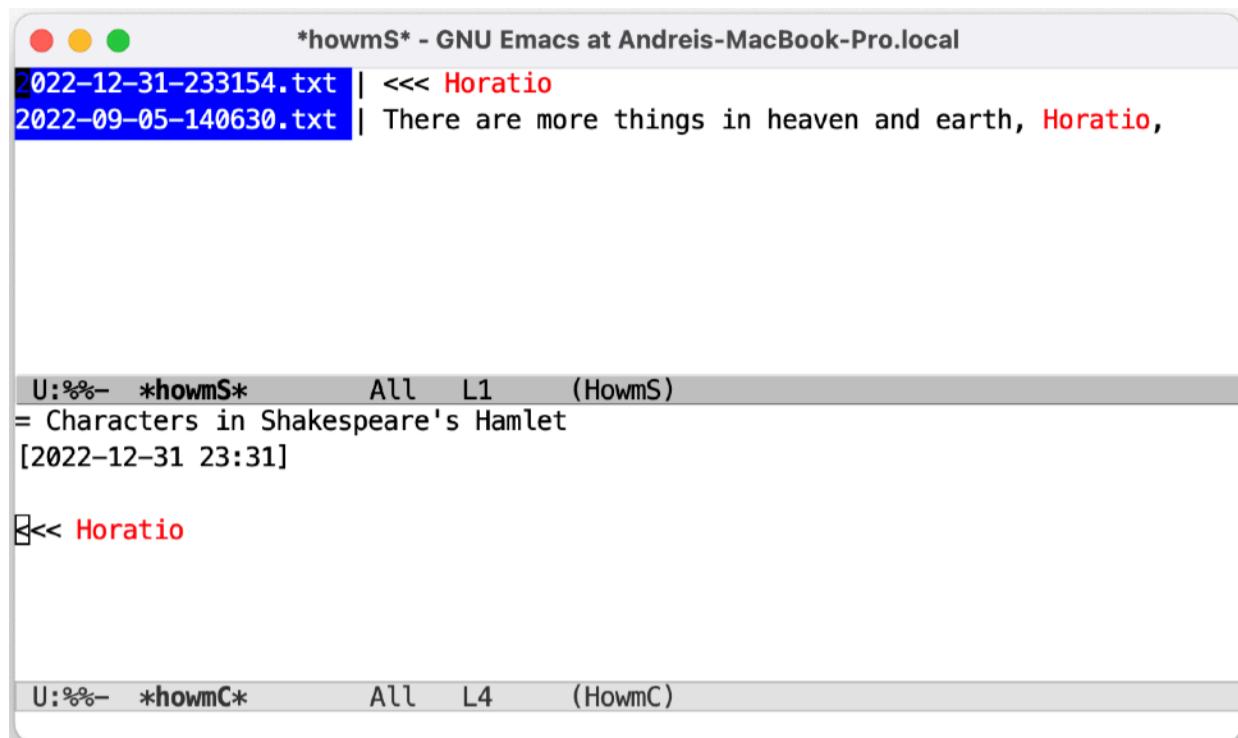
The spruce pointing to the left <<< has the exact opposite meaning. You don’t want to go anywhere; you want all other notes with the keyword to lead here. This is “come-from link.”

Anything after the <<< sign will be treated as a link. So, in order to continue with regular text, you must start a new line.

Let’s take a look at how this link works with a concrete example.



If you place the cursor on a link in Note A and press RET, you will be forwarded to the standard two-window screen with the list on top.



Once again, let me remind you that the link serves as the trigger for the search query, and the intermediate result is a list of notes that match the criteria<sup>38</sup>.

Since the note with the “come-from link” is always at the top of the list, you only have to press RET to jump to it. Thus, by placing the cursor on the underlined word and pressing RET twice, you are immediately taken to the address note.

### Three important remarks

1. “Goto link” is activated as soon as you type it. This does not happen with a “come-from link.” You need to save the file `C-x C-s` for it to work.
2. When you type a keyword in a new note, it is not dynamically converted to a link. Howm needs to report changes to the text. To do this, you can save the note or press the command `C-c , r`. This command can also be run from the howm’s menu (where it is the `r` key).
3. “Come-from links” are case sensitive. If you capitalize a word, the same word but with a lowercase letter in other notes is not converted to a link. However, search is not case-sensitive. Therefore, all variants will appear in the search results (intermediate step).

“Come-from link” can be useful in creating different kinds of indexes (subject, topic, person, source).

---

<sup>38</sup> More on search functions in the next chapter.

This type of links in howm have one more interesting and beneficial feature. You can specify an alias for a keyword, simply put, a list of synonyms or similar concepts in a string.

<<< heuristics <<< serendipity <<< intuition

Once you have created a chain like this, all the specified terms in other notes will lead to this particular memo. The search list will combine the results for all three words.

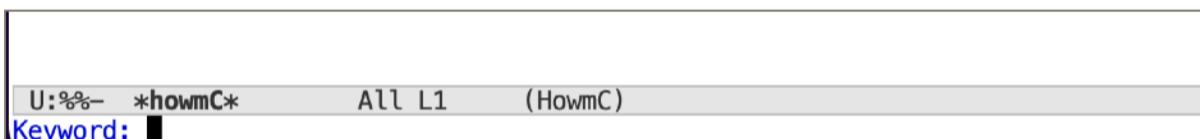
“Come-from links” can perform a similar role to tags.



What if you don't want all mentions of a word, but only specific notes? Use some sort of sign in front of the word, and then howm will search for the specified variant. For example, this link <<< :intuition: will ignore ordinary mentions of the word “intuition” in other notes. If you work with some documents where tags are indicated via a hash, use this: <<< #intuition.

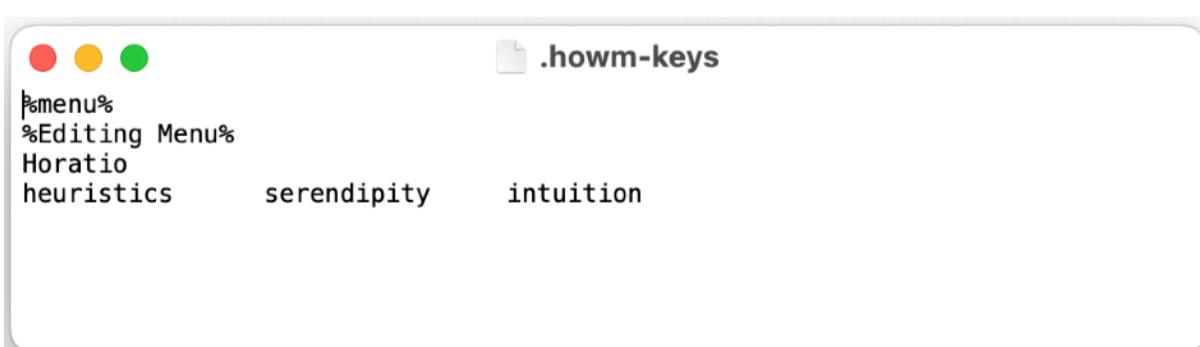
Of course, remembering keywords and re-entering them is a cumbersome task. At a certain stage of work there will be quite a lot of them. Howm has a special command for calling all keywords - C-c , i

The command gives you access to a list of all keywords entered in the database. To do this, howm calls the minibuffer dialog:



You can type the first few letters of the word you are looking for and then press the tab key (in howm, as everywhere else in Emacs, autocomplete works). If you want to see the full list of keywords, press the tab key at once.

It is worth clarifying here that howm puts all the keywords in a special file called .howm-keys



If you haven't changed anything in the settings, it is located right in your home directory. Although it is better to place it in the `howm` folder<sup>39</sup>.

By the way, you can edit this file directly. A word entered in this list is under `howm`'s supervision, even if it is not mentioned in any memo as a "come-from link."

This can be handy, especially in the case of tags. Clicking on a link-tag opens a list of tagged notes, sorted in the order of the newness of edits.



Keep in mind that simply deleting a "come-from link" will not deactivate links in other notes. They will still trigger a search, but the list will now be without the address note at the top. To remove a "come-from link" completely, you must delete the corresponding word in the `.howm-keys` file.

I still haven't mentioned an additional type of links that can be used in `howm`. These are the square bracketed links familiar to many people from wiki and markdown. Here, they behave in a specific way.

The screenshot shows two Emacs buffers. The left buffer, titled "2023-02-14-224214.txt", contains text in Greek and English. It includes a note about self-knowledge and a maxim about self-acceptance. The right buffer, titled "2023-12-07-000055.txt", shows a link from the self-acceptance note back to the original note. Red arrows highlight the creation of the link and its automatic heading creation.

**2023-02-14-224214.txt - GNU Emacs at Andreis-MacBook-Pro.local**

= Know thyself  
[2023-02-14 22:42]

σωντὸν ἴσθι  
Know thyself

This is the most famous of the Delphic maxims.

But in order to succeed, we must also remember another maxim:

σεαυτὸν εὖ ποίει  
Treat yourself well

It's easy for self-knowledge to slip into self-gnawing, which undermines the whole process.

[[Self-acceptance]] Is the centerpiece of successful self-knowledge.

U:\*\*- 2023-02-14-224214.txt All L16 (Text Fly/- howm AL Wrap)

Create a name for the new note and enclose it in square brackets. This activates the link. Place the cursor on the link and press RET. This will create a new note.

A link to the note "Know thyself." Howm adds such links whenever one note is created from another.

The note heading is created automatically. It is a "come-from link," so any mentions of the term "Self-acceptance" will now lead to this note.

The term itself is included in the `.howm-keys` file.

2023-12-07-000055.txt - GNU Emacs at Andreis-MacBook-Pro.local

= << Self-acceptance >>  
[2023-12-07 00:00] >> ~/howm/2023-02-14-224214.txt

U:\*\*- 2023-12-07-000055.txt All L1 (Text Fly/- howm AL Wrap)

Wrote /Users/Sukhovskii/howm/.howm-history

<sup>39</sup> See "Advanced Settings," p. 68

Advantages of square bracketed links:

1. You create a link, and howm, when you click, creates a note (if it doesn't already exist).
2. The link does not require it's own line, as in the case of spruce-links. As a result, the text looks more compact.

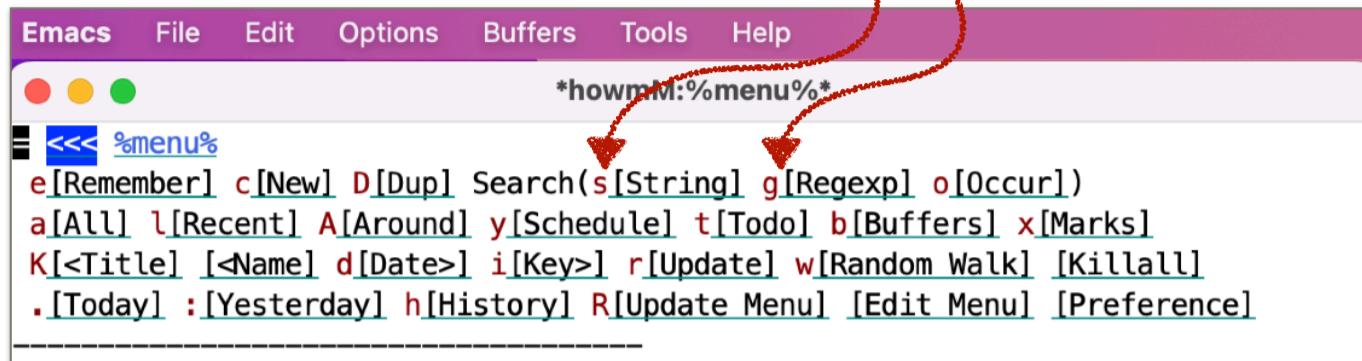
Disadvantages of such links:

1. Such links only work well with note headings; however, numeric links are more reliable and versatile.
2. Note headings will be accumulated in a .howm-keys file, which is wisely reserved for indexes and tags. I don't know if this will affect system performance, but the keyword search will definitely be more difficult.

# SEARCH FUNCTIONS

In this section, we will look at the search functions in howm. There are many useful tricks here that allow you to navigate efficiently.

The howm's menu contains the necessary hints.



To initiate a search, move the cursor to the appropriate menu item and press RET or immediately press the key associated with the command.

If you are already working with notes, there is no need to go back to the initial howm page<sup>40</sup>.

The search is invoked by keyboard shortcuts: `C-c , s` и `C-c , g`

The `C-c , s` command allows you to search for any text fragment, ranging from a part of a word to a complete phrase.

Additionally, thanks to this command, you can access the list of keywords entered into the howm database (the `.howm-keys` file). If you press the tab key after invoking the command, you will receive their complete list. Type the first letters of the desired term and press the tab key again for autocomplete<sup>41</sup>.

In this case, the result will be the same as if you used a “come-from link” in some memo (remember that clicking on a link in howm is just a search query?).

The `C-c , g` command also allows you to search for any text fragments. But here the search functionality is extended by regular expressions.

For example, by writing: Wozle \l Wizzle you will get all the notes that contain these words. This is called a logical OR.

<sup>40</sup> You can also initiate the search immediately after starting Emacs

<sup>41</sup> I've already mentioned the Vertico package. There is a side effect when using it in this case. Sometimes Vertico is complaisant. Let's say you have “philology” in your keyword list. If you try to find notes with the word “philologist,” you won't succeed. The initial letters in your search query match the word in the list, and Vertico will autocomplete the missing letters. In a such situation, use the command `C-c , g`

There may be more intricate commands. If you wish to explore them, consider reviewing the regular expression syntax (regexp). These features are optional for working in howm.

In the case of a simple search query, you can use shortcuts: `C-c , s` и `C-c , g` interchangeably.

Let's now examine how howm behaves during the search process.

Query any word, and howm will provide you with a list of notes that contain it (the search by default is case-insensitive).

In the minibuffer, next to the search query, you will see the number of matches in your knowledge base indicated as [n hits]. After typing a word, press RET.

The screenshot shows the Emacs interface with two windows. The top window, titled '\*howmS\* - GNU Emacs at Andreis-MacBook-Pro.local', displays a list of note files and their excerpts. The cursor is over the file '2023-08-19-222953.txt'. The preview window below shows the contents of this note, which discusses the 'Pluralism Fallacy' and 'Godwin's law'. The minibuffer at the bottom shows the search term 'The Pluralism Fallacy' and the date '[2023-08-19 22:29]'. The status bar indicates 'U:%%- \*howmC\*' and 'View: /Users/Sukhovskii/howm/2023/12/2023-08-19-222953.txt'.

```
*howmS* - GNU Emacs at Andreis-MacBook-Pro.local
2023-08-19-222953.txt | = The Pluralism Fallacy
2023-12-07-133526.txt | Let's call it the "novelty advantage fallacy."
2023-11-14-193607.txt | Well, meditation develops a kind of psychical strength $
2023-03-20-132822.txt | Meditation cannot help directly in correcting cognitive$
2023-12-07-132901.txt | In many discussions, when participants feel that their $
| Now this is known as "Godwin's law". This is of course $
| Okay, now we see that "Reductio ad Hitlerum" is a falla$
| We know that "Argumentum ad hominem", "Argument from ignoran$
| Nevertheless we need to avoid one misunderstanding. Some people $
```

U:%%- \*howmS\* All L1 (HowmS)

☰ The Pluralism Fallacy  
[2023-08-19 22:29]

If you are familiar with internet debates, you might notice that there are a variety of rhetorical devices that participants use. Sadly, not all of them are legitimate.

Often it is nothing more than intellectual deception. Some of these devices are developed to persuade an opponent; some to attack the opponent's argumentation even by blocking further discussion.

The latter includes the phrase "There is more than one point of view." Of course, this phrase is not always a trick. Eventually, almost every conversation consists of different opinions

U:%%- \*howmC\* Top L1 (HowmC)

View: /Users/Sukhovskii/howm/2023/12/2023-08-19-222953.txt

The search output is divided into two windows. The upper one contains a list of notes with the word you are searching for. The lower one previews the note on which the cursor is placed, allowing you to quickly scroll through the notes.

The top window is more important for us now. Here, we have a blue bar with numeric names of notes, and to the right of it, excerpts from notes where the word we are looking for is highlighted in red.

If a word occurs more than once in a note, all occurrences will be listed, each on a separate line. In the left column, repeated mentions are easy to spot. It is an empty dark blue cell, indicating that the mention is in the same note whose numeric name is given above.

You can collapse these additional lines by pressing the `u` key. This way, only the first mention of the word in each note will remain in the list. To return to the detailed summary, press `u` again.

Sometimes, it is easier to navigate through the material by note titles rather than excerpts. Press the `T` key, and howm will display them to the right of the numeric name. This provides an additional tooltip to help you find the material you're looking for more quickly. To return to the list of excerpts, press the `T` key again.

A search result with a list of notes is a good place to start. However, in howm, you can further refine and reorganize the data you've got.

Sometimes, a single-word search creates too large a list of notes. For instance, you may not want all the notes on philosophy. Instead, you are interested in notes that mention both philosophy and Aristotle. This is called a logical AND.

In howm, such a query has to be performed sequentially. First, search for one word, and then narrow the search for another.

There is an upside to this. You can conduct one round of searching and then decide how to narrow it down. You can repeat the operation again and again.

This function is tied to the `G` key. After searching for the first word, press `G` and enter the second word.

You can filter out notes from the search result in other ways as well. More on this in the next chapter.

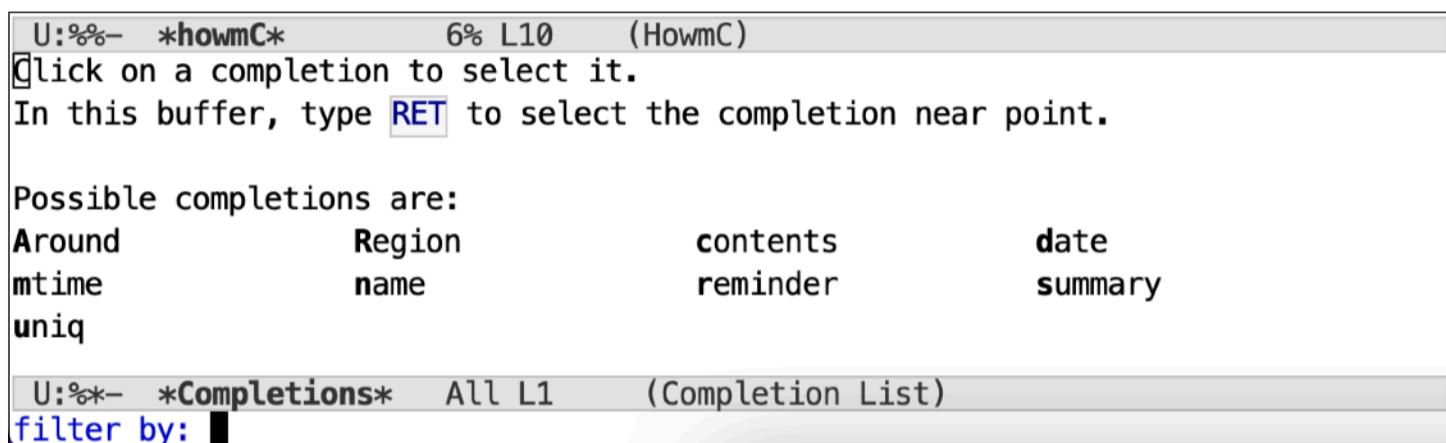
One more useful feature I will mention here. Sometimes you have to search for the same memos frequently. Howm has a search history that makes life easier in such cases. It is called by the keyboard shortcut - `C-c , h`

# FILTER FUNCTIONS

Too much information? Narrow it down.

In any list press the **f** key, and howm will open the dialog in Minibuffer. The filters it provides can be used to specify search criteria or request particular collections of notes.

There are plenty of options here to help you focus on an exact segment of information. Press Tab to see the full list.



Let's now consider the available options in order of importance.

## **name**

This is the numeric file name in the blue column. Specifying 2023-03, for example, will give you a list of notes created in March'23. However, if you want notes that you created last Monday, then add the date to the year and month.

## **date**

A filter similar to the previous one. However, here you can specify a range. For instance, you might be interested in notes created between December 2022 and February 2023.

## **mtime**

Here you can also specify a time range, but the output will be a list of notes that you have worked with during the specified period (i.e., not only created but also modified).

## **summary**

This filter allows you to narrow down the number of notes by their title. For instance, you only want notes that have the name “Aristotle” in the headings.

You can also filter search results. As you remember, to the right of the blue column with the numeric file name is an excerpt with the search word highlighted. The excerpt includes the line in which the word is located. You can narrow your search results further, asking for other words that occur within the same line (say, Socrates and Plato).

## contents

With this command `howm` will search inside the already selected notes. This command performs the same function as the `G` key. It is useful when you have a list of notes with the word you are looking for and then decide to narrow it down further.

## reminder

A command that allows you to filter task lists if you use them in `howm`<sup>42</sup>.

## Region

In Emacs, you can select text or list items by pressing `C-SPC` and then moving the cursor. Highlight the notes you want, then press `f` and select the Region option. `Howm` will remove everything else from the list.

## Around

A filter that allows you to take any note (by moving the cursor over it) and query a list of other notes you have worked with before and after the specified one. This is useful when you need to recall context.

You can specify the number of notes in a context. For example, if you want to display a context of 10 notes before and 10 notes after the active note, press `C-u 10` before the `f` key and choose “Around.”

## uniq

A command that duplicates a function bound to the `u` key.

\* \* \*

You can shorten the list of notes not only by using a logical AND but also with a logical NOT.

For example, you might want to exclude notes about Kierkegaard from a search on philosophy. Use the keyboard shortcut `C-u f`. `Howm` will prompt you to specify the criteria for elimination.

---

<sup>42</sup> We will discuss task management in one of the following chapters.

You have a choice of...

#### **name**

A numeric file name (you can cross out one specific note, or you can cross out an entire month or year).

#### **date**

Same culling by date, but you can specify which date range you want to remove.

#### **mtime**

Culling by the date of edits. For example, you want to remove from the list notes on philosophy that you worked with a year ago.

#### **summary**

You can cross out notes based on the excerpt to the left of the numeric file name or based on the title (toggle with the **T** key).

#### **contents**

Cross out notes based on content.

The **Around**, **Region**, **reminder**, and **uniq** functions are useless in this case, so don't pay attention to them.

All of these clever tricks for sorting and filtering information allow you to create a workspace with only the notes you need. This also helps with further work on articles and essays.

Unfortunately, howm doesn't have an option to save all these tricky filter combinations for reuse. But on the other hand, I realize that I would create a lot of filters that I would get confused with.

# DELETING NOTES

Sometimes a note needs to be deleted. You should do this only in very rare cases. For example, you have discovered that entry A is a complete duplicate of entry B.

There is no quick delete method (one click) in howm. Maybe it's for the best. Amputation of texts requires awareness.

First, check if the note you want to get rid of is linked to other records. Careless deletion may result in broken links in the system.

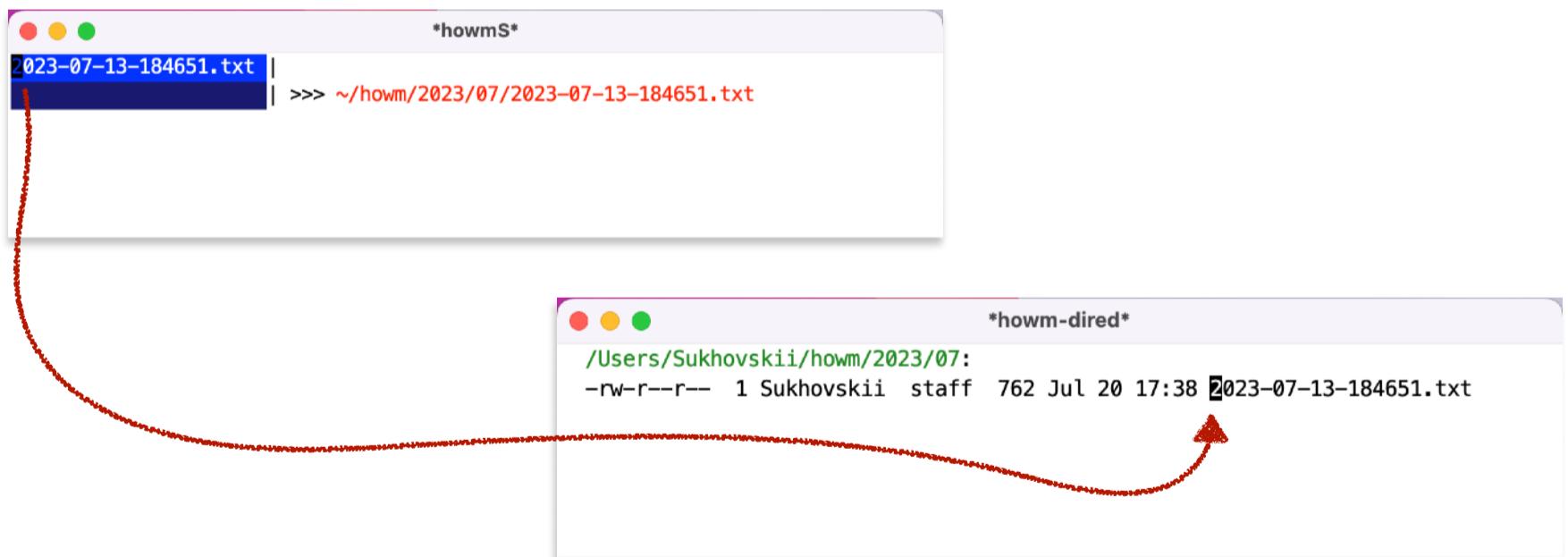
The easiest way to check is as follows:

Use self-reference trick in the memo<sup>43</sup>. Move the cursor to the link and press RET. Only one note should appear in the list.

Using hyperlinks on headings instead of numeric names makes the procedure almost the same. Copying heading is done with the command `C-c , K`

Once you are confident that note removal will be harmless to the knowledge base, you can proceed with the surgical procedure.

To do this, switch to Dired mode. If the cursor is positioned on a targeted note, press `X`. The screen will change, but the cursor will still point to the same note.



In the new window, you have access to all features of Dired mode<sup>44</sup>.

<sup>43</sup> We've already discussed how to do this with the example of backlinks. See p. 30

<sup>44</sup> Both here and elsewhere, familiarity with the Emacs documentation will significantly enhance your freedom of action.

Press `v` to look in the file if reassurance is needed. Press `q` to return to the Dired mode.

To delete a file, press `D`. Emacs will ask you if you are aware that the action is irreversible, and after your consent, deletion will be completed<sup>45</sup>.

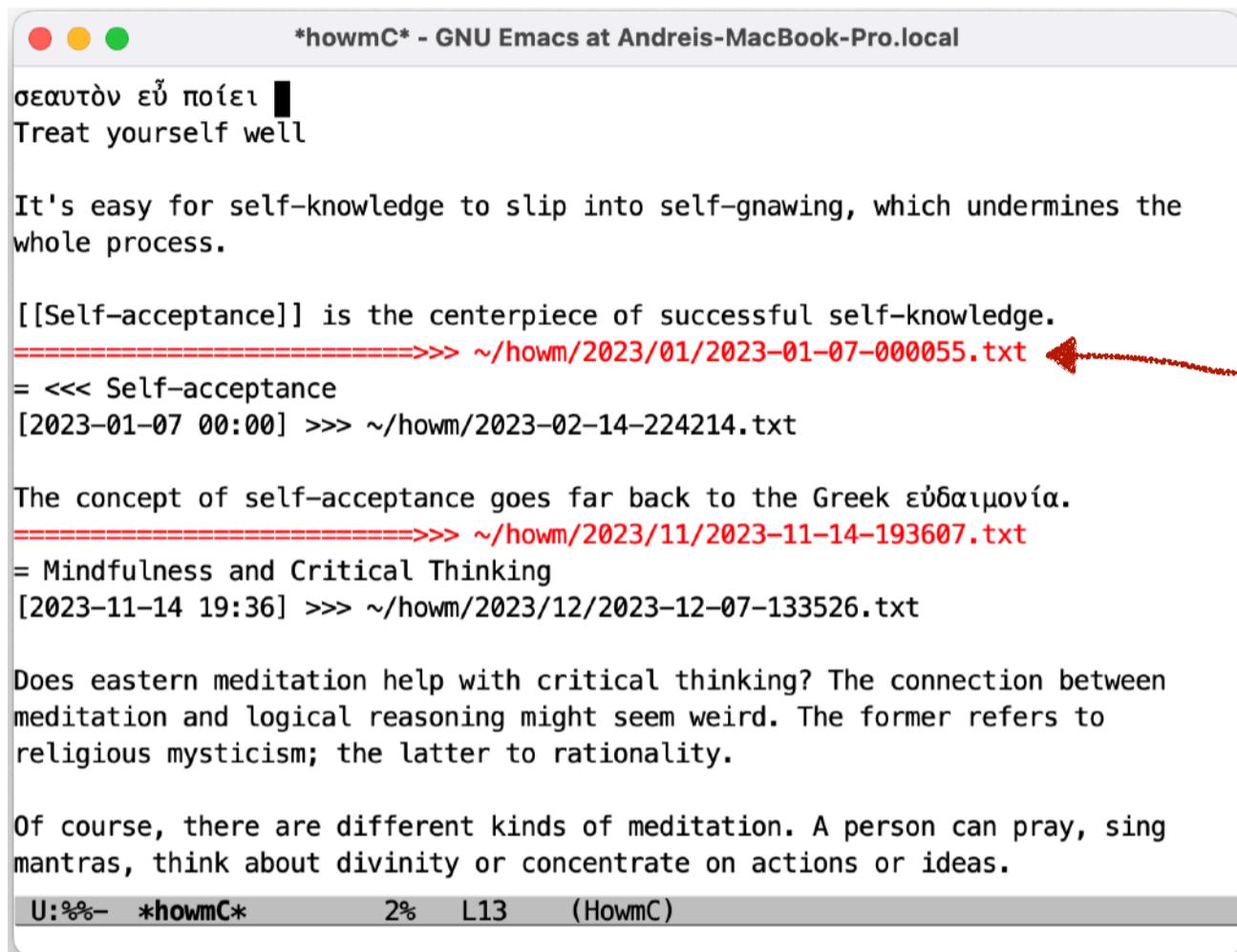
You can select several files for deletion at once. To do this, press the `d` key in sequence. Emacs will mark the files. Press `x` to confirm the operation. If you decide to unmark a file, move the cursor to the required line and press `u`.

---

<sup>45</sup> The file does not go to the bin but is completely erased from the system, so be careful!

# STITCHING NOTES

In howm, it is possible to read notes in a merged form, as a concatenation. Notes will be temporarily stitched together into a single scroll with markers indicating the start of a new note.



The screenshot shows a terminal window titled "\*howmC\* - GNU Emacs at Andreis-MacBook-Pro.local". The window displays several concatenated notes:

- σεαυτὸν εὖ ποίει  
Treat yourself well
- It's easy for self-knowledge to slip into self-gnawing, which undermines the whole process.
- [[Self-acceptance]] is the centerpiece of successful self-knowledge.  
=====>> ~/howm/2023/01/2023-01-07-000055.txt
- = <<< Self-acceptance  
[2023-01-07 00:00] >>> ~/howm/2023-02-14-224214.txt
- The concept of self-acceptance goes far back to the Greek εὐδαιμονία.  
=====>> ~/howm/2023/11/2023-11-14-193607.txt
- = Mindfulness and Critical Thinking  
[2023-11-14 19:36] >>> ~/howm/2023/12/2023-12-07-133526.txt
- Does eastern meditation help with critical thinking? The connection between meditation and logical reasoning might seem weird. The former refers to religious mysticism; the latter to rationality.
- Of course, there are different kinds of meditation. A person can pray, sing mantras, think about divinity or concentrate on actions or ideas.

At the bottom of the window, the status bar shows: U:%%- \*howmC\* 2% L13 (HowmC)

To switch to the concatenation view of any list of notes, type `0` or `@` key. You can exit the view by pressing one of the specified keys again.

This feature has several practical applications.

Sometimes, you really need to overcome the discreteness of the notes collection and see the records in a single text. It's much more convenient to read them this way. This is in keeping with howm's motto: "Write fragmentarily and read collectively."

You can jump from note to note using the `Tab` and `M-Tab` keys.

The concatenation function is also useful for quickly searching and reviewing notes. Kazuyuki Hiraoka suggests using the following algorithm:

1. Conduct an initial search query: `C-c , s`
2. Merge the result into a single scroll.
3. Use the text search to find the information you need: `C-s`

In addition, concatenation can serve as a stage from dispersed memos to drafts of articles.

The algorithm may be as follows:

1. Put a special label on the entries you want to include in the article. This can be any unique combination of letters, for example - art23.
2. When you're done marking notes, use the search by the dedicated label. This will form a list.
3. Arrange the notes in the list in the logical order you want. To do this, you will have to use this trick:
  - A. Press the `S` key to display the sorting dialog.
  - B. Use the `name-match` or `summary-match` option to move notes around. The reverse sequence should be used here.
    - i. Find the first note and move it to the top of the list.
    - ii. Find the second note and move it to the top of the list.
    - iii. Repeat the procedure all the way down to the last note.
    - iv. Press `R` to reverse the list.
4. Call up concatenation view - `0`
5. Select the text with `C-x h` and copy it with `M-w`.
6. Open a text editor and paste the copied fragment<sup>46</sup>.

Since notes have now taken on a second life in the article, it is helpful to register this fact in the knowledge base. This practice will help you avoid self-repetition. Replace art23 label with “used in so-and-so.”

If you don't have a lot of notes, you can do it manually, but Emacs has a great Search and Replace feature for such tasks.

Press `C-x d` and navigate to the directory where you want to perform the operation. In our case, it is the `~/howm/`

Press the `t` key. This will mark all files in the directory. Next, press `Q`. A dialog will appear in the minibuffer.

---

<sup>46</sup> Of course, if you use Org-mode in Emacs to work with long texts, all these tricks are unnecessary. You can skip step 3 and rearrange notes in Org's outliner.

Type art23 and press RET. Specify a replacement - “used in article...” and press RET again.

Emacs will ask you how to do the replacement. Press the `Y` key to apply it in all files at once (the label is unique, and we don't have to worry about some word being substituted by mistake).

Now you need to save the files that have been modified. To do this, use the standard `C-x s` command and then confirm by clicking on the exclamation mark (saving all files at once).



You can get a report on the total number of words and characters in your knowledge base. To do this, “stitch” all the notes into a single text.

1. Call up the list of all notes - `C-c , a`
2. Press the `0` or `@` key.
3. Enter `M-x count-words` and press RET.

In the minibuffer, you will see how much you have written. However, the figure will not be entirely accurate because howm includes special markup for dividing notes in the list. But in general, the error is not significant.

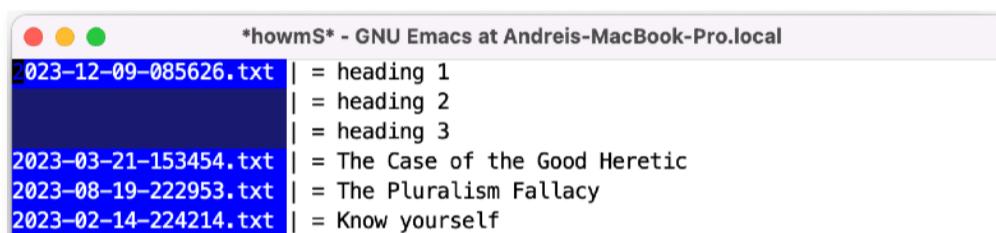
# HANDLING LONG NOTES

In some cases, you may require a long text rather than a short entry. If lengthening cannot be avoided, it is wise to at least break up the material with separate headings.

Each note in howm starts with a heading. However, you can include additional sections in the text:

```
= heading 1  
= heading 2  
= heading 3
```

If you call up a list of notes, for example, using `C-c`, a shortcut, it will look like this:

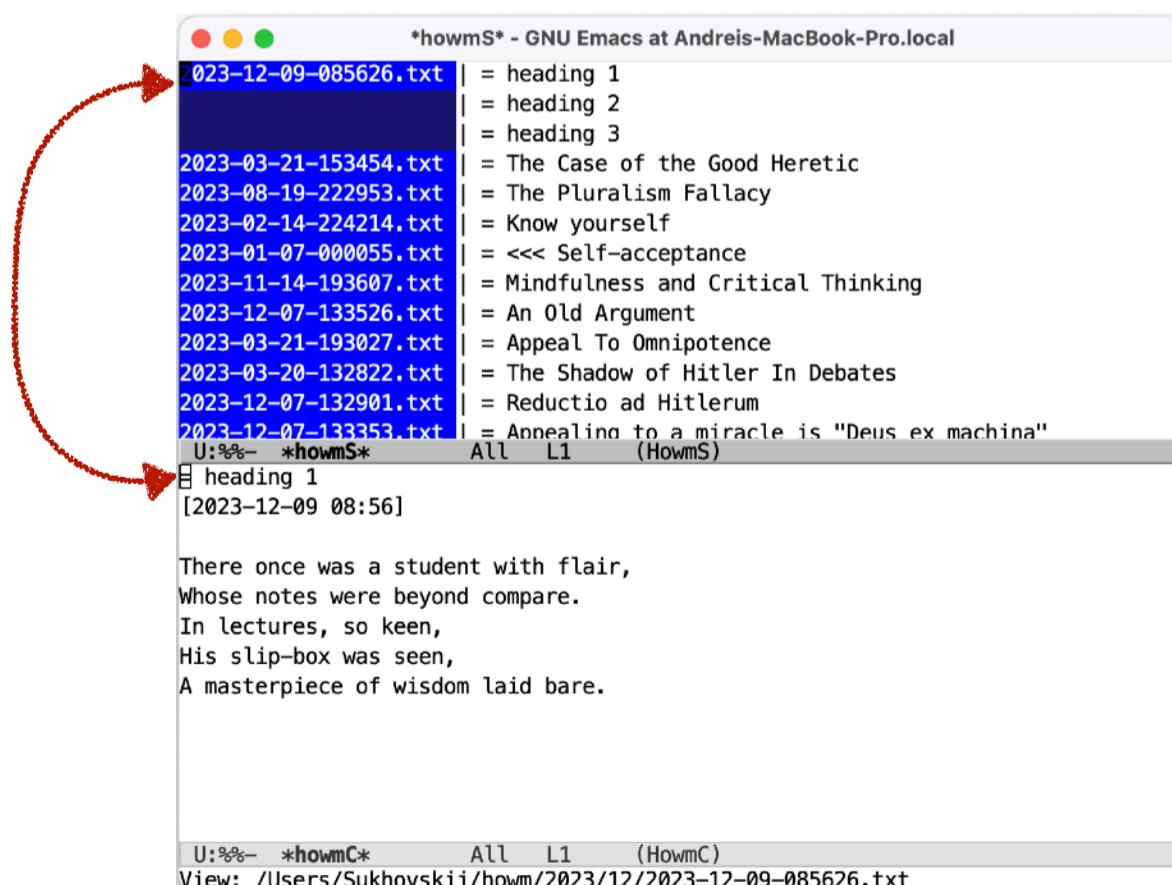


\*howmS\* - GNU Emacs at Andreis-MacBook-Pro.local

Note Name	Headings
2023-12-09-085626.txt	= heading 1 = heading 2 = heading 3
2023-03-21-153454.txt	= The Case of the Good Heretic
2023-08-19-222953.txt	= The Pluralism Fallacy
2023-02-14-224214.txt	= Know yourself

On the left in the blue column is the note's numeric name, and on the right are all the headings the note contains.

Notice that as you move the cursor, howm will show each heading in the preview window as a separate note (that is, entries under other headings will not be visible until you move to the note).



\*howmS\* - GNU Emacs at Andreis-MacBook-Pro.local

2023-12-09-085626.txt | = heading 1  
| = heading 2  
| = heading 3  
2023-03-21-153454.txt | = The Case of the Good Heretic  
2023-08-19-222953.txt | = The Pluralism Fallacy  
2023-02-14-224214.txt | = Know yourself  
2023-01-07-000055.txt | = <<< Self-acceptance  
| = Mindfulness and Critical Thinking  
2023-11-14-193607.txt | = An Old Argument  
2023-12-07-133526.txt | = Appeal To Omnipotence  
2023-03-21-193027.txt | = The Shadow of Hitler In Debates  
2023-03-20-132822.txt | = Reductio ad Hitlerum  
2023-12-07-132901.txt | = Annealing to a miracle is "Deus ex machina"  
2023-12-07-133353.txt | = There once was a student with flair,  
| Whose notes were beyond compare.  
| In lectures, so keen,  
| His slip-box was seen,  
| A masterpiece of wisdom laid bare.  
U:%%- \*howmS\* All L1 (HowmS)

heading 1  
[2023-12-09 08:56]

There once was a student with flair,  
Whose notes were beyond compare.  
In lectures, so keen,  
His slip-box was seen,  
A masterpiece of wisdom laid bare.

U:%%- \*howmC\* All L1 (HowmC)  
View: /Users/Sukhovskii/howm/2023/12/2023-12-09-085626.txt

This will work in a similar way with search results. You can search for a note by a specific title. Then that title will appear as a separate note in the upper window<sup>47</sup>.

The screenshot shows a split-screen Emacs window. The top half is titled '\*howmS\*' and contains the text 'heading 2'. The bottom half is titled 'View: /Users/Sukhovskii/howm/2023/12/2023-12-09-085626.txt' and contains the full poem:

```
There once was a man, quite aged,  
Whose slip-box was like a wise sage.  
With cards of all kinds,  
His knowledge aligned,  
In a one mile well-ordered page.
```

Naturally, clicking on RET and jumping into a note will reveal the title in the context of other notes.

You can also use the equals sign for footnotes:

```
= Heading  
Lorem ipsum dolor sit amet...  
=  
footnote 1  
=  
fotnote 2
```

In this case, howm will show the first line of the footnote in the upper window (right column). In the lower window, the full text of the corresponding footnote will be displayed.

If a note has grown into a long text with many headings, you need to navigate quickly. You can jump from heading to heading using the following commands:

C-c , N → Next heading.

C-c , P → Previous heading.

C-c , H → Go to the first heading.

C-c , L → Go to the last heading.

---

<sup>47</sup> In the main menu, in the Recent section, each title will also be presented as a separate note. The numeric name will naturally be the same.

To create a new section in a note, simply start a line with an equals sign. Next, you can add a title if necessary. But you can also use a template to insert a heading with a time code: `C-c , C.` The new heading is appended to the end of the text regardless of where your cursor was.

For long notes, the function of focusing on a specific part is useful. Howm does not have a separate keyboard shortcut for this, but the feature itself is available.

The `M-x howm-narrow-to-memo` command will hide all parts of the text except the current one (on the title of which the cursor is positioned)<sup>48</sup>.

Before executing the command, howm will clarify your intentions. This is because this function is disabled by default.

In the minibuffer, you will see the following options:

`y` - activate for this and following sessions.

`n` - don't activate.

`SPC` - activate the function once. In this case, howm will execute the command, but the function will remain locked.

`!` - activate for the current session.

By hiding other sections, you can work with the desired piece of text as if it were a separate note.

To return to the display of all headings with texts, press `C-x n w`

You can toggle between hiding and unhiding other sections using the command:

`M-x howm-toggle-narrow`

\* \* \*

If the note is extensive, you may need to search for a particular passage in the text. The simplest way is to use the standard `C-s` command.

What if you want a list of all the matches? Howm has a special feature for this.

---

<sup>48</sup> The beauty of Emacs is that you are quite capable of assigning your own keyboard shortcut to invoke a command. For example, write the following in your `.emacs`:

```
(define-key howm-mode-map (kbd "C-c , =") 'howm-narrow-to-memo)
```

Now you will be able to narrow the text to a specific section with the `C-c , =` command.

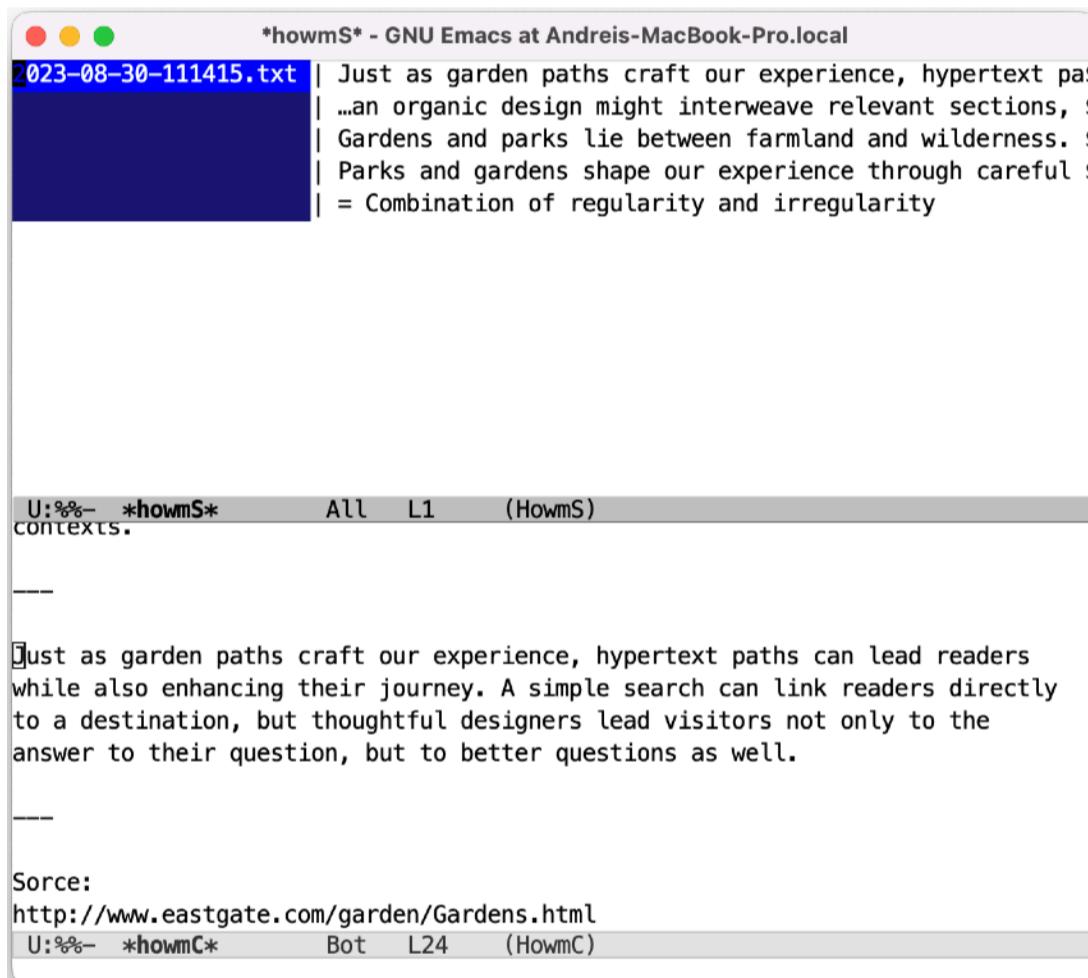
While in a note, type `C-c , o`, input your query, and you'll access the standard howm interface: a list at the top window and a preview at the bottom. But in this case, you will navigate within a single note.

Another technique that makes it easier to work with long notes is the use of marks.

To create a mark, press `C-SPC C-SPC`

When the marks are assigned, you can call the list of them with the shortcut command `C-c , x`

Moving between such “Marginalia” is much faster than normal swiping<sup>49</sup>.



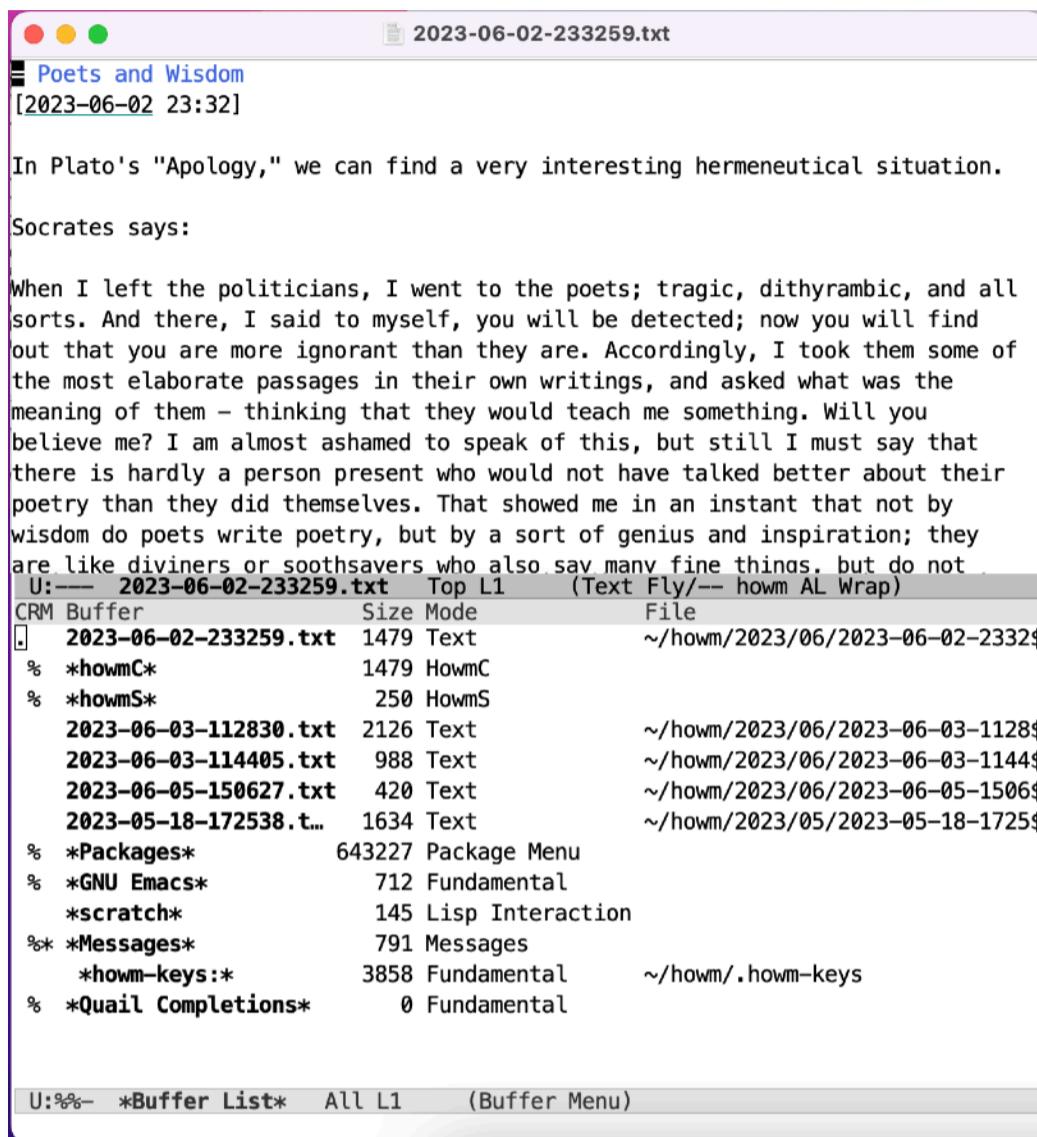
Since a link in howm is a search query trigger, you can use special labels to refer to a specific passage in a long text. For example, in one note, you would place the label - qq, and in another, the link >>> qq. When you click on the link, howm will open the document in the desired location<sup>50</sup>.

<sup>49</sup> Howm opens the marks of the active note. Marks in other files you visited during the session are not lost, but to view them, you need to navigate to the corresponding memo.

<sup>50</sup> Of course, it is advisable to develop a system of unique labels if you will actively use this technique. The Q-trick was proposed by Merlin Mann in 2010. He used it when working with notes in Simplenote and Notational Velocity.

# WORKING WITH BUFFERS

Emacs stores every document you open in a special buffer list. The standard Emacs command to call the list is `C-x C-b`



The buffer list allows you to accomplish many different tasks. It is better to refer to the official Emacs documentation<sup>51</sup>.

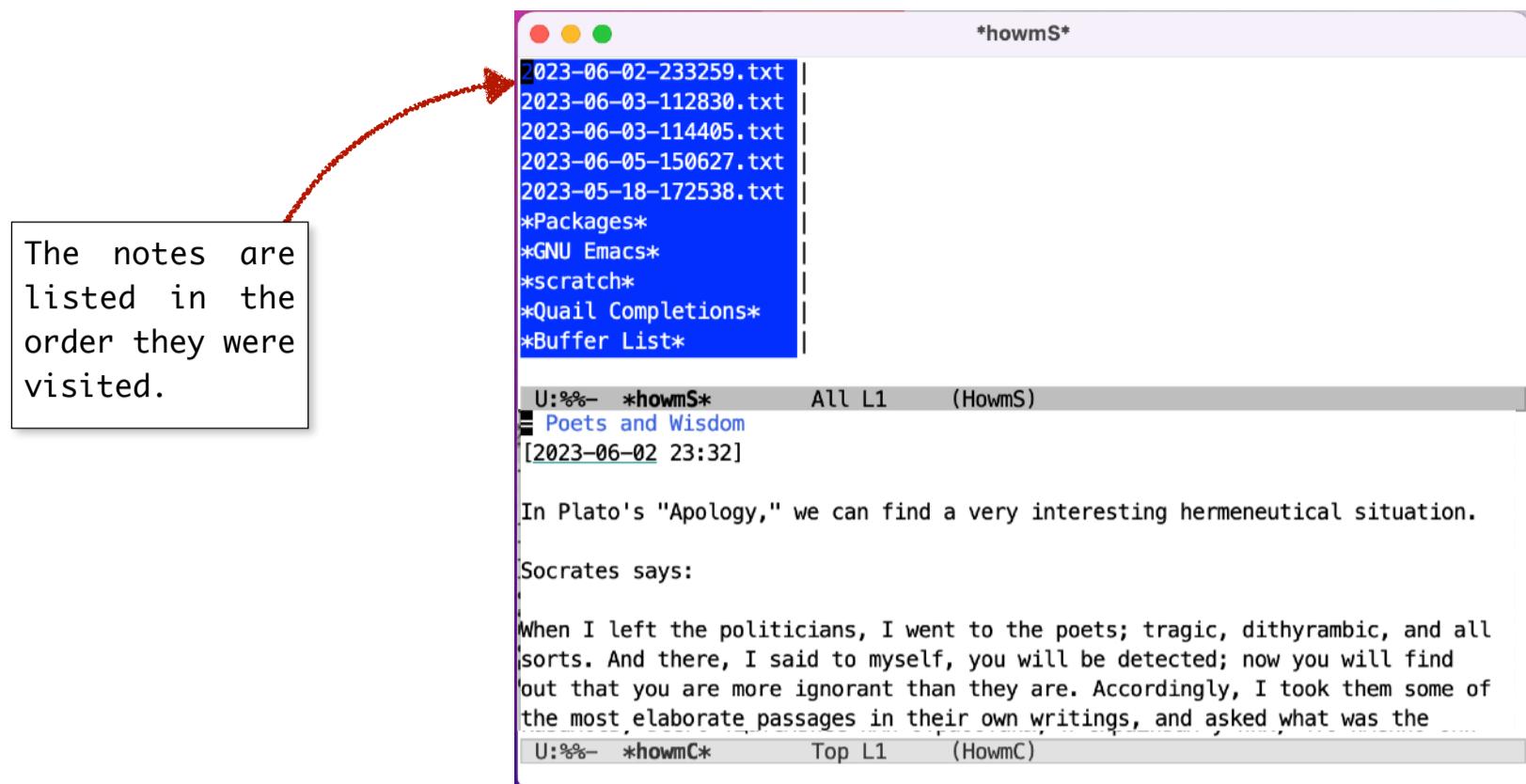
You can switch between notes in howm and other documents using the `C-c , SPC` command. This will take you to the last document that was opened outside howm mode. Typing the command again will bring you back to the last memo opened in howm<sup>52</sup>.

Howm also has its own command for calling the buffers list - `C-c , b`. The advantage of this command is that you get the howm's double screen. The upper window lists the buffers, and the lower one previews contents.

Other documents opened in Emacs will also be available.

<sup>51</sup> Check it out here: [https://www.gnu.org/software/emacs/manual/html\\_node/emacs/Buffers.html](https://www.gnu.org/software/emacs/manual/html_node/emacs/Buffers.html)

<sup>52</sup> It doesn't work smoothly with all modes. You will be able to switch between documents in txt format. However, if you have opened a document in Org-mode, the howm commands will not function. In this case just use `C-x <` go back. See also "Advanced settings," p. 71



This feature is handy to use as “breadcrumbs,” i.e., to track your path through the knowledge base. By checking the list, you can find the files you have visited during the current session<sup>53</sup>.

As you work on your notes, you will accumulate a lot of open buffers. Some of them still need to be at hand, while others are getting in the way. How to clear the space?

You can type the command - `C-c , Q`

This will close all but the unsaved howm buffers (howm will ask you if you want to do this). You will then be taken to the last unsaved howm file or, if there are none, to the last open buffer not associated with howm<sup>54</sup>.

If you have already called up the list of buffers - `C-c , b` - just press `Q`

The command to close buffers is also available from the main menu - [Killall]

---

<sup>53</sup> Numeric file names are displayed in the list. Using titles would be more intuitive. Fortunately, improvements can be made accordingly. Refer to the “Advanced Settings” section, p. 70 for more details.

<sup>54</sup> The standard Emacs command for closing the buffer - `C-x k` - works too.

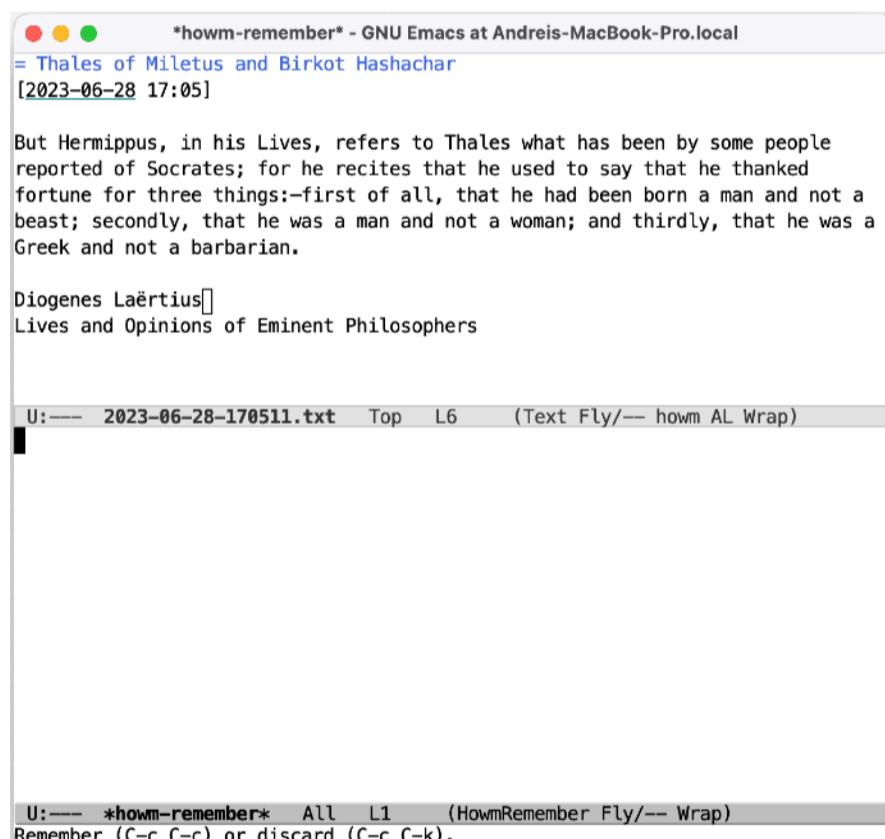
# QUICK NOTES

Sometimes, while writing one entry, an idea for another comes to mind. You don't want to interrupt the flow state, but you also don't want to miss the insight.

Howm has a quick note function for such situations. The purpose of it is to create a simple prompt on a topic you want to develop later without getting distracted for a long time. In practice, it is just an empty note with a title and date. Something you want to remember.

In the main menu, the e[Remember] button is responsible for this function. However, you can call up the quick note buffer from anywhere by typing `C-c , e`

Howm will open an additional completely blank window. There will be command hints in the minibuffer.



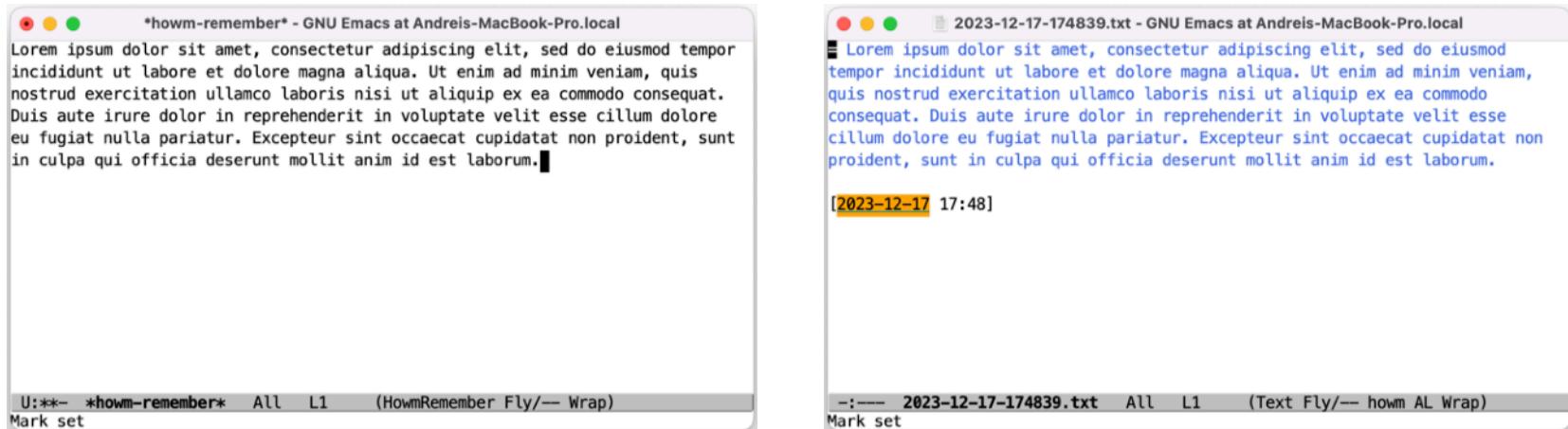
Ideas placed here haven't been registered in the knowledge base yet. If you decide they're unwanted, press `C-c C-k`, and Howm will close (kill) the buffer, saving nothing<sup>55</sup>.

To move a note to the knowledge base, press `C-c C-c`. Howm will create a file with a numeric name and kill the buffer. You will return to the note you were working on before.

When it's time to revise quick notes, you can call up the list of recent files - `C-c , l` - or open the main menu.

<sup>55</sup> When Emacs closes, the contents of this buffer are lost. Emacs will not ask whether to keep it, as it does with other documents. This operates similarly to the \*scratch\* buffer.

The quick notes feature is designed for entries that are one sentence long, a headline. If you have a whole paragraph instead of a sentence, the template will become messy.



Of course, you can reconfigure this feature to create expanded drafts. However, you can only do this by changing the settings globally. Enter the following line in the .emacs file:

```
(setq howm-template "= %title\n%date %file\n\n%cursor")
```

Now the text of a quick note will be placed in the body instead of the heading (It can be composed later). However, when creating a regular note, the cursor will also not be in the title field but below the date.

\* \* \*

One more specific thing about the quick notes template is worth noting. By default, a blank line is added between the title and date in square brackets. This is not a bug but a legacy. The blank line plays some role in packages unknown to me. You can reconfigure this function so that the templates of regular and quick notes are the same<sup>56</sup>.

---

<sup>56</sup> Refer to “Advanced settings,” p. 67-68 for details.

# DUPLICATION

For some types of notes, having a template is useful. This helps avoid countless repetitions of the same activities. For example, in a bio cards, we will need the following columns: years of life, interests, etc.

In howm this task can be accomplished by using the duplicate function. The command `C-c , D` creates a clone of the current note. The duplicate is no different from the original, except for the new numeric name and the fact that the first line is empty (it is left for the new title).

This action can also be called from the main menu - `D[Dup]`. In this case the last open buffer will be duplicated.

In notes with multiple headings (see p. 49), only the section of text where the cursor is located is duplicated.

This actually opens up some interesting possibilities.

First, you can create a single document with a collection of templates separated by headings.

Second, you can easily duplicate a part of any note by simply adding the equal sign above and below the selected fragment.



A document with a template can be titled with a special code, such as - `T_01` (so they can be quickly called through search).

If you changed your mind about making a duplicate after you clicked the command, that's okay. Click `C-/` and close the window. No traces will be left in the system<sup>57</sup>.

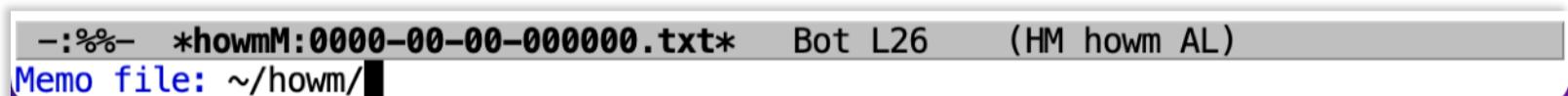
---

<sup>57</sup> This is similar to canceling when creating a note (see p. 17).

# DESCRIPTIVE NAMES

I already mentioned that you can move any plain text files (txt, md, org, tex) to the `howm` folder. And they can be named whatever you want. Of course, consistency is better for the system to work properly. Numeric names are good for their universality and abstractness. Nevertheless, a file named `graphomania.txt` will work in the system without any problems.

You can also create a file with a descriptive name in `howm`. The `C-c , I` command is reserved for this purpose. It opens a special dialog box.



Type the desired file name and extension.

If you want files with descriptive names to be located in a separate folder, you need to create it beforehand (it can also have any name). The best way to do this is to use the standard Emacs command:

1. `M-x make-directory RET`
2. Type `~/home/chosen name RET`

Once the directory is created, you are free to place any files there.



Howm treats all files placed in its directory as part of the knowledge base, regardless of which sub-directory they are in. In all listings, they will be presented together.

Once again, I'll note that you have to have a serious reason for abandoning the standard `howm` naming system.

By the way, you can view any text file outside the `howm` directory by using the command: `C-c , M`

If you want the `howm` syntax to work as well, type:

`M-x howm-show-buffer-as-howm`

# TASK MANAGEMENT

When it comes to personal knowledge management, it is immediately clear that this activity implies at least minimal planning. Some books need to be read by a certain deadline, some information needs to be entered into a system, some material needs to be revised, an article needs to be re-read, etc.

If the knowledge base becomes an everyday tool, it makes sense to manage your intellectual work (or even all your tasks) right there. Howm suggests a system of flexible scheduling designed to help disorganized people. The idea is to turn chaos to your advantage instead of fruitlessly trying to eradicate it.

A key element of working with information in howm is lists. They are highly suitable for planning. However, any to-do list tends to grow to a size incompatible with life expectancy. You need to ensure that not all tasks are visible at once.

The principle here is the same as in organizing notes. Wasting time on categorization proves to be counterproductive. Hiraoka writes that he took the drastic step of abolishing “importance” and instead implemented a list based on “seasonality.” This metaphor signifies that reminders will catch your eye at the appropriate time.

You have a few tools at your disposal. You can use lists with schedules, deadlines, todo, reminders, and deferrals<sup>58</sup>.

Hiraoka also employs the imagery of tasks sinking (moving to the bottom of the list) and tasks surfacing (moving to the top of the list). This enables you to easily identify what is currently relevant, as it will always be at the top of the list. The unnecessary sinks, while the essential ones stay afloat.

Let’s get into how all this “hydromechanics” works.

A quick overview of the plans is possible in the main menu of howm. Here, a list of todo, reminders, etc. is generated. Howm extracts this data from the notes, so you don’t have to worry about where you wrote it down. It won’t get lost.

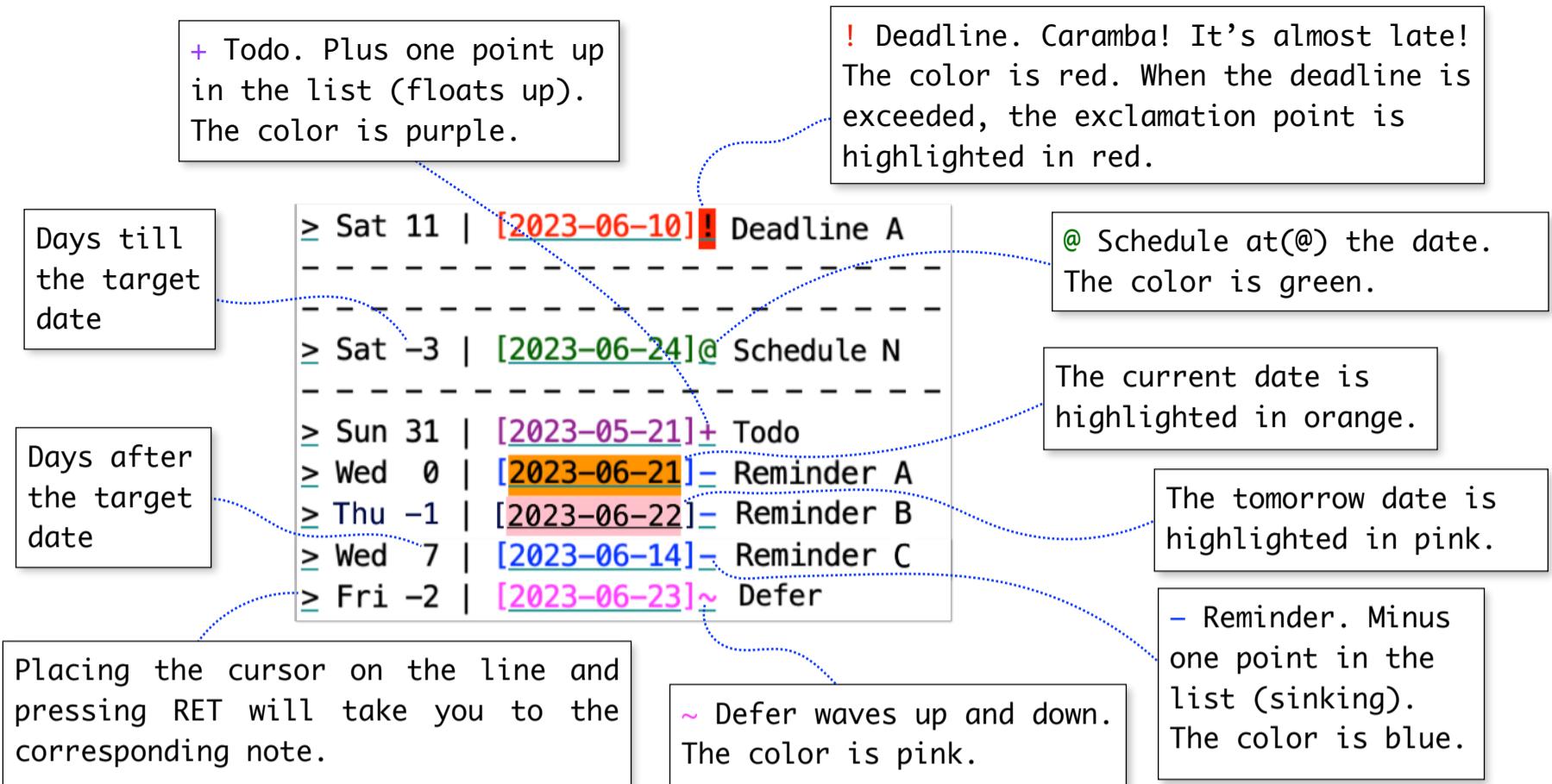
The list in the main menu is not exhaustive; it has a depth of 50 items and covers a seven-day schedule. But that’s the point: to keep out of sight the tasks that belong to another “season.”

There is a special syntax for denoting different types of plans in howm, with mnemonic cues.

---

<sup>58</sup> A little further down, we’ll take a look at what all of this looks like in howm.

## Planner Syntax



The **scheduled** event appears in the list seven days before the appointed date and instantly surfaces to the top of the list. Overdue events are removed from the list the next day. Quite sensible, as you don't need a reminder of a concert that took place yesterday.

The **deadline** is at the bottom of the list. Seven days before the date, it surfaces to the top and stays afloat even after the deadline.

**Todo** is at the bottom of the list. It starts to gradually float to the top of the list seven days before the due date.

The **reminder** is at the bottom of the list. It appears at the top on the appointed date and then sinks a little each day, eventually diving to the bottom of the list again<sup>59</sup>.

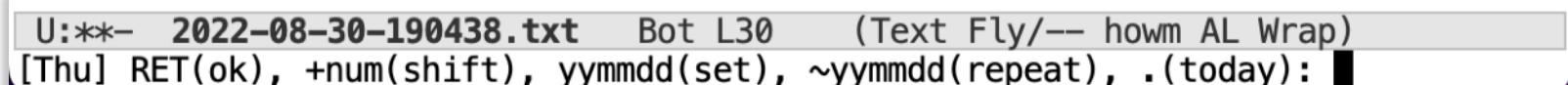
The **deferred** behaves like a reminder. The difference is that resurfacing and diving are cyclical. It will float up over and over again at 30-day intervals.

Whatever type of plan you have, it will be bound to a date.

Date format: [YYYY-MM-DD]. You can enter this data manually, but howm has its own clever ways of doing this.

<sup>59</sup> The number of dive points will be determined by the dates of other todos and reminders.

The command `C-c , d` will insert the desired date into an open note<sup>60</sup>. When executing this command, howm will open a dialog in the minibuffer:



U:\*\*- 2022-08-30-190438.txt Bot L30 (Text Fly/-- howm AL Wrap)  
[Thu] RET(ok), +num(shift), yymmdd(set), ~yymmdd(repeat), .(today): █

For now, let's focus on a simple task: if you need today's date, you can simply press either the dot or just RET.

What if you have plans for January 5, 2026? Just dial 260105 and press RET.

In most cases, there's no need to provide the full date; Howm understands various abbreviations.

Numbers from 1 to 31 will be interpreted as the date of the current month<sup>61</sup>. The number 803 will be construed as August 3 of the current year.

By typing `+14` and then pressing RET, you will add fourteen days to today's date.

The `C-c , T` command will insert today's date and time without additional questions in the format: `[yyyy-mm-dd HH:MM]`

If you type `{_}` and then press RET with the cursor inside, the effect is identical<sup>62</sup>.

Either way, you are now a master of precise calendar events. What comes next? Following the date, you need to include the type of plan and provide an explanation. For example: `[2023-12-30]!` Complete the annual report.

You can create a designated note with your plans or organize them within different notes. In a single note, you can list tasks consecutively or intersperse them with comments—it doesn't matter. Howm enables you to navigate swiftly through any variety of tasks. If you save a file with a task and go to the main menu, you will see that it is included in the summary list and coordinated with other plans.

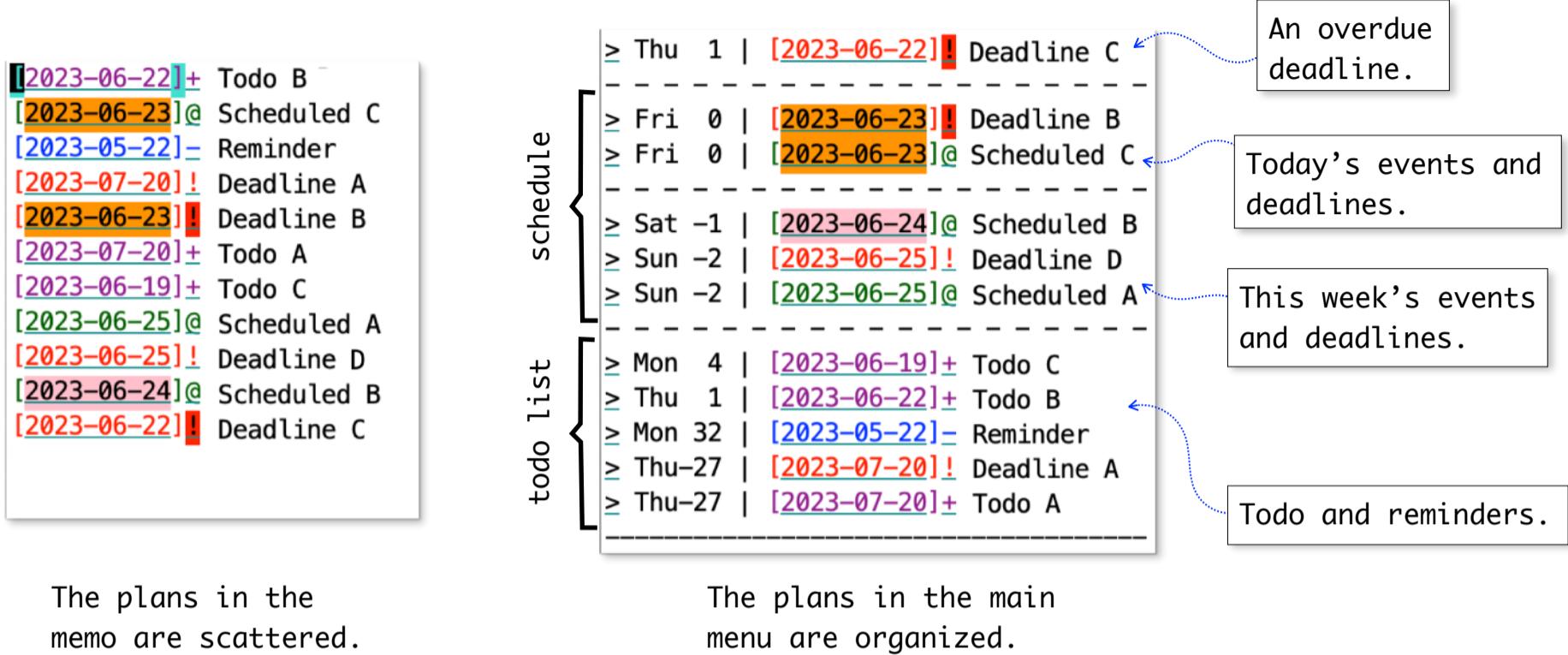
---

<sup>60</sup> The howm menu features a key labeled `d[Date]`, which inserts a date into the last opened note (the last buffer). However, the date will be inserted where the cursor was located. If you haven't opened any notes before executing the command, it will not function as expected.

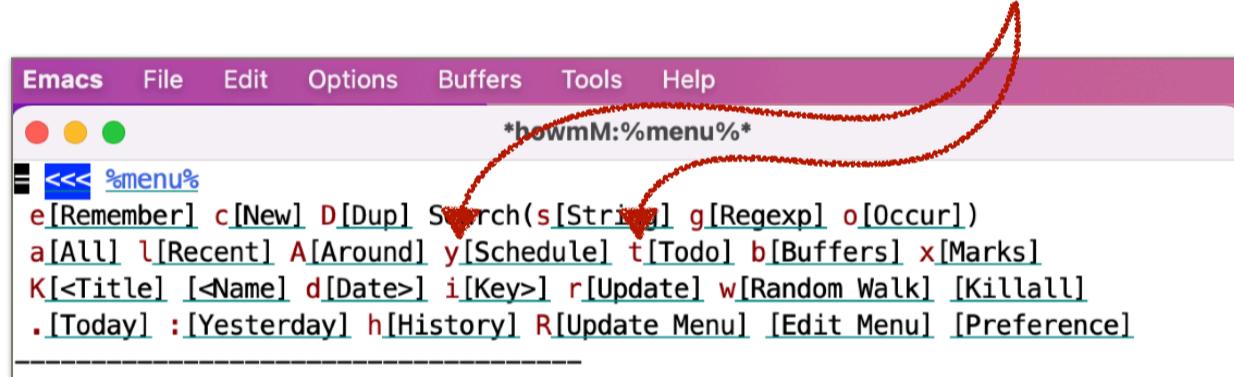
<sup>61</sup> If it's June and you type 31, howm will interpret it as 30+1 and consider it to be the first of July.

<sup>62</sup> This is an example of a function called action-lock. It can automate various processes in howm (refer to "Advanced Settings," p. 71 for more details).

Todos, reminders, and deferrals will be listed together in the main menu. Scheduled events will appear in a separate block. Depending on the date, deadlines may be included in the list with todos or placed in the schedule block<sup>63</sup>.



You can access the list of todos and events not only in the main menu. Howm provides two special commands that enable you to work with plans.



The todo list can be called from anywhere with the command `C-c , t`. You will get the same interface as you are used to when working with notes. The advantage, especially when tasks are scattered across different files, is that you can quickly preview what is planned.

The list of scheduled events is called with the command – `C-c , y`

Deadlines will appear on both todo list and scheduled events list, ensuring it's impossible to miss them.

Unlike the main menu, lists are not limited in number of items. You have access to all plans<sup>64</sup>.

<sup>63</sup> The schedule displays events and deadlines in the order of dates. In contrast, reminders and todo are shown in a “seasonal” manner.

<sup>64</sup> E.g. events scheduled for the next quarter will not be visible from the main menu.

Certainly, with numerous plans, it's easy to get lost. Simply tap on a dot, and the cursor will move to the activities or events appointed for today.

Todo lists and events are arranged based on their "seasonality." If you wish to view them in calendar sequence, utilize the sorting function. Open a list, press **S**, and choose the reminder option. Howm will then rearrange the list to align with the dates.



You can integrate plans hosted in howm with Emacs calendar packages. One package worth exploring is `emacs-calfw`, available at <https://github.com/kiwanami/emacs-calfw>

A good planning system should be flexible. Everything flows, as Heraclitus said. Circumstances change, we change... We must have a means of quickly adjusting goals.

To change the date, position the cursor inside the square brackets and press RET. Howm will then open the dialog box, as previously mentioned on page 61.

To simply postpone a task, for example, by four days, type `+4` and press RET. You can also reschedule the task for today by pressing a dot, or you can specify a number between 1 and 31, interpreted as a date (the month remains unchanged). If you want to change both the day and the month, specify the date in MDD format, e.g. `901` for the first of September.

You can reassign the date not only from the note but also from the main menu. The actions are the same: move the cursor inside the square brackets with the date and press RET. Howm will temporarily take you to the corresponding file with an open dialog in the minibuffer. Once the operation is complete, you will automatically return to the menu<sup>65</sup>.

At times, you may need to modify the type of event rather than the date. For instance, a reminder can be transformed into a deadline, a scheduled event into a reminder, and so forth.

Position the cursor to the character after the square brackets and press RET. A dialog will appear in the minibuffer. Enter the character (`~@-!+`) corresponding to the new status and press RET.

```
U:--- 2022-08-30-190438.txt Bot L41 (Text Fly/-- howm AL Wrap)
RET (done), x (cancel), symbol (type), num (laziness): █
```

<sup>65</sup> Keep in mind that the changes have not yet been saved. To do this without leaving the menu, use the command **C-x s**

Similar to dates, you can make changes to records directly from the main howm menu<sup>66</sup>.

In addition to changing the event type, you can set parameters for its behaviour (quickness of surfacing/sinking in the list). To do this, you need to specify a number after the symbol (~@-!+). In the minibuffer, which we looked at on the previous page, there is an option: num (laziness). All you have to do is specify the desired digit and press RET. The higher the number, the slower the downward movement and the faster the upward movement.

Let's look at some examples.

By default, a scheduled event disappears from the radar the day after the appointed date. However, by specifying [2023-06-25]@5, you can make the event stay afloat for five more days.

Deadlines surface at the top of the list 7 days before the date. If you want a deadline warning 20 days in advance, type [2023-06-25]!20

The reminder floats to the top by the appointed date, then sinks. The rate of sinking is adjustable. For example, a reminder like [2023-06-25]-9 will sink every nine days. Number 999 will keep reminder afloat for years.

The todo gradually starts to float to the surface of the list 7 days before the date. If you want it to start surfacing earlier, increase the number, for example: [2023-06-25]+14

For a deferred task, the number represents the length of the sinking-surfacing cycle. If you wish to see an entry at the top of the list more than once every 30 days, specify this, for example: [2023-06-25]~15

It's time to discuss the changing status of plans. Sooner or later, there will come a moment to fulfil or abandon a plan.

Position the cursor on the symbol after the square bracket and press RET. The minibuffer dialog (refer to the previous page) offers two options.

If you press RET again, the case will be considered complete, and its display will change.

[2023-06-04]+ attain enlightenment



[2023-06-28]. [2023-06-04]:+ attain enlightenment

---

<sup>66</sup> Certainly, when editing plans within the note itself, it's easy enough to erase the character and type in a new one—the number of clicks remains the same.

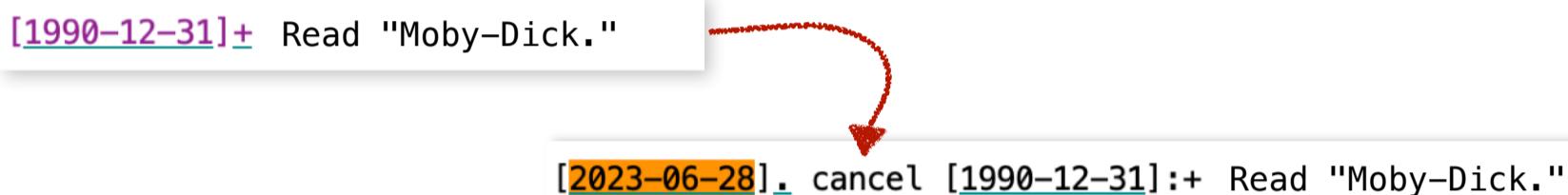
The sign of the end of the case is a period. Automatically, howm adds the date of the finale. This way, you have a handy tool to analyze your progress.

However, you can use the simplified syntax as well. Place a notation somewhere: [2023-06-28]. attain enlightenment

That way you will know that the case is at an end.

The other possible outcome is that the plan is canceled. It's okay not to keep up with everything.

While in the minibuffer dialog, press the `x` key



The notation “cancel” will remind you that the task is not completed but overruled. Otherwise, the syntax remains the same.

Completed and canceled plans are submerged to the bottom of the list. The main menu doesn't have them at all.

 You can view the completed plans with the commands `C-c , t` or `C-c , y`

As completed cases are at the bottom, press `R` to turn the list upside down.

The last feature to consider in this section is repetitive tasks.

Howm only has the option of a limited number of repetitions. This approach has its pros and cons.

The cons are obvious. You cannot, for example, create a cyclic task once for all<sup>67</sup>. You won't be able to determine what important event will fall on the second Monday of December next year (although cases where information about highly delayed events is needed are rare).

The pros of limited repetition is that you consciously decide how many times to take an action, whether to continue, etc.

<sup>67</sup> Notice, however, that deferral is cyclical.

To create a recurring task/event, position the cursor inside the square brackets and press RET. The familiar minibuffer dialog will then appear in front of you:

```
U:**- 2022-08-30-190438.txt Bot L30 (Text Fly/-- howm AL Wrap)
[Thu] RET(ok), +num(shift), yymmdd(set), ~yymmdd(repeat), .(today): █
```

Specify the end date of the cycle. For example: ~250601

Then, press RET, and howm will prompt you to specify the desired repetition rate. There are a few options here.

```
U:**- 2022-08-30-190438.txt Bot L31 (Text Fly/-- howm AL Wrap)
Every? [RET(all), num(days), w(week), m(month), y(year)] █
```

**RET** - Daily repeat until the specified date.

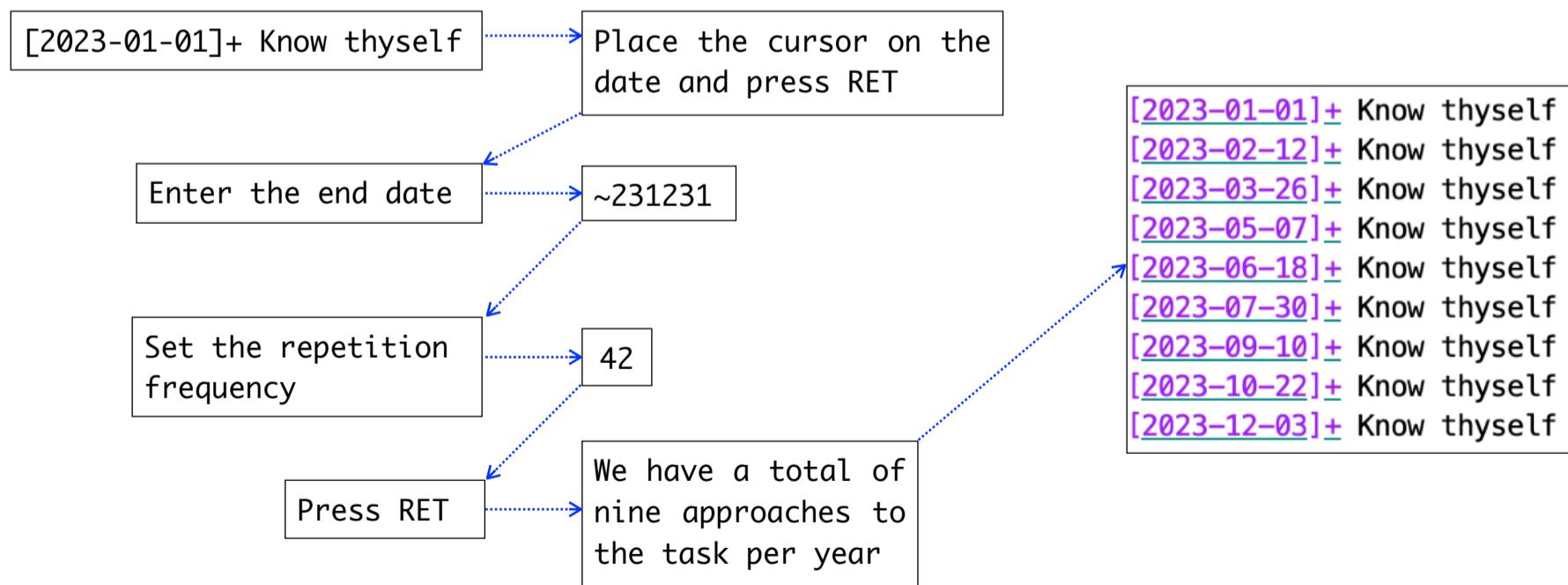
**num** - How many days between repeats.

**w** - Weekly, on the same day of the week as the start date.

**m** - Monthly, on the same date of month as the start date.

**y** - Annually, the same month and date as the start date.

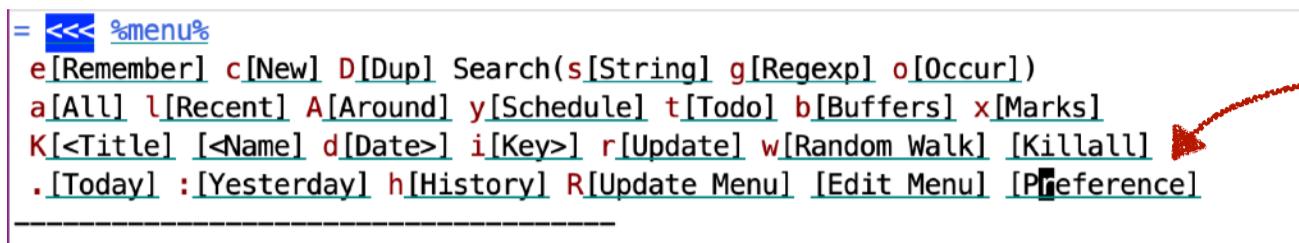
Example:



Howm clones the task a specified number of times, changing the date. The advantage is that you can then reassign the date of an individual item without breaking the rest of the chain (roughly the same way it works in org-mode).

# ADVANCED SETTINGS

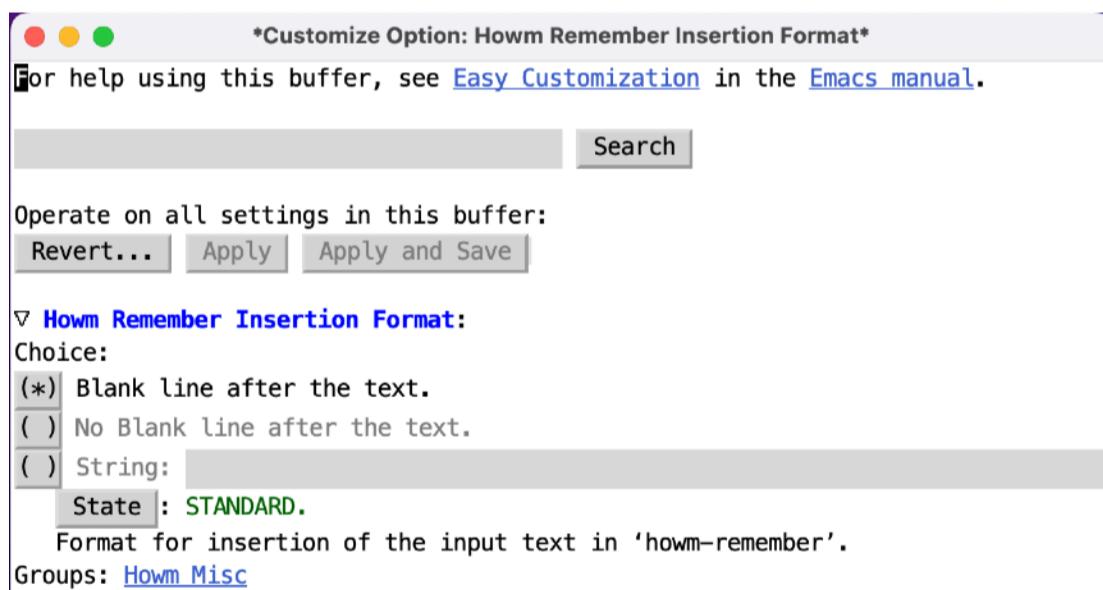
You can access customization section directly from the main menu<sup>68</sup>.



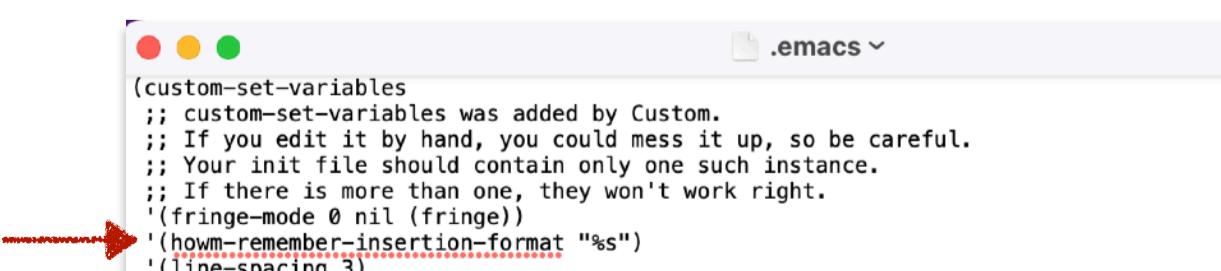
Here you can change any of the values. Make sure to click **Apply and Save** for the changes to take effect. Your preferences will be written in the .emacs file under the **custom-set-variables** section (where the document starts).

If you reconsider, changes can be undone by going to the appropriate section of the **howm** group and clicking **State** button. In a dialog, you will see an option **Erase Customisation**. Click on it and then on **Apply and Save**. Alternatively you can simply remove the corresponding lines from **custom-set-variables** in the .emacs file. Howm will forget about all modifications.

Let's look at a concrete example. Suppose you want to remove a blank line after the title in a quick notes template. Go to **Preference** → **Howm Misc** → **Howm-remember-insertion-format**. Select the second option: No Blank Line after the text



After that, click on **Apply and Save**. Now, if you check the .emacs file, you will find a new line.



<sup>68</sup> Alternate path: M-x customize → Applications → howm

Power users often resort to directly editing the `.emacs` file. You could also write the appropriate line there:

```
(setq howm-remember-insertion-format "%s")
```

However, keep in mind that you will not be able to undo such changes through the settings. You will have to go back into `.emacs` to make the corrections manually<sup>69</sup>.

\* \* \*

Since it's not unreasonable to be skilled in working directly with `.emacs`, we'll explore advanced customization options below. There is nothing particularly complicated about them; it's just a code that tells `howm` what to do and how to do it.



To keep the system files neatly packed in the `howm` folder in your home directory, add the following to your `.emacs` file:

```
(setq howm-home-directory "~/howm/")
(setq howm-directory "~/howm/")
(setq howm-keyword-file (expand-file-name ".howm-keys" howm-home-directory))
(setq howm-history-file (expand-file-name ".howm-history" howm-home-directory))
```



If you want to save files in a specific format (markdown in the example), add the following:

```
(setq howm-file-name-format "%Y/%m/%Y-%m-%d-%H%M%S.md")
```



The menu file (`0000-00-00-000000.txt`) is located in the same folder as the notes. As a result, it is listed along with other entries. If you find this bothersome, you can move it outside of the `howm` folder. Don't forget to specify the new location in `.emacs`:

```
(setq howm-menu-file("~/path-to-the-file/0000-00-00-000000.txt"))
```

---

<sup>69</sup> If you make some changes through the settings panel and others directly, conflicts may arise. It is advisable to keep a log of changes for yourself to avoid confusion.



You can specify additional directories in which howm should search for notes.

```
(setq howm-search-path '("/path/to/notes/directory" "/path/to/other/notes/directory"))
(setq howm-search-other-dir t)
```



If you want to integrate Howm with the email client under Emacs, add the following lines to your .emacs file (be sure to specify the appropriate file path):

```
;; search ~/Mail and ~/News in addition to normal notes
(setq howm-search-path '("~/Mail" "~/News"))

;; initial value of M-x howm-toggle-search-other-dir
(setq howm-search-other-dir t)
```



Another useful customization pertains to the viewing of the only-list screen.

```
(setq howm-view-keep-one-window t)
```

It disables the bottom preview window while scrolling through a long list of notes.



When the number of notes exceeds a thousand, Howm becomes a bit pensive (not terribly) when loading the main menu. You can optimize the process by adding the following lines to your .emacs file:

```
(setq howm-view-use-grep t) ;; use external grep
(setq howm-menu-refresh-after-save nil)
(setq howm-menu-expiry-hours 1) ;; cache menu N hours
(setq howm-menu-file "0000-00-00-000000.txt") ;; don't *search*
```

It may not save you much time, but the experience will be smoother.

There are side effects! The menu will stop updating automatically while you revisit it. But it's not a big deal. There is a command in the menu itself to update the content - R[Update Menu]. Another side effect is that the search in howm becomes case sensitive.



“Come-from” links are case-sensitive. If a word in .howm-keys is capitalized, the same lowercase word will not be underlined in the notes. However, if you follow the link, all mentions of the word will be listed, regardless of case (case-insensitive search). If you want all key-words, regardless of case, to be underlined, add the following line to your .emacs file:

```
(setq howm-keyword-case-fold-search t)
```

Note that if you have set up a search involving grep (see above), the function will not work correctly.



When viewing a list of open buffers, the numeric names are uninformative. The following two lines, added to your .emacs file, solve this problem. Instead of the numeric names, you will see the note titles in the list. This will make navigation much easier. Of course, in reality, the numeric names will remain intact; only the display will change.

```
(add-hook 'howm-mode-hook 'howm-mode-set-buffer-name)  
(add-hook 'after-save-hook 'howm-mode-set-buffer-name)
```

In long notes with multiple headings, this setting starts to behave strangely. Sometimes the numeric name of the buffer is replaced not by the first heading, but by the second or third.



If you want Howm to create one file for each day, add this line to your .emacs configuration:

```
(setq howm-file-name-format "%Y/%m/%Y_%m_%d.txt") ; one file per day
```

In this case, howm will append new entries to an already created note, providing dated headings. It will work as long note (see pp. 49-52).



When you press RET on a special symbol (!+-~@), howm opens a dialog asking for the next action. You can make it so that pressing RET means that the task is finished. To do this, add the following line to your .emacs file:

```
(setq howm-action-lock-reminder-done-default "")
```

In this case, the dialog will be called with the command - C-u RET



If you open an org document outside the howm directory, keyboard shortcuts will not work. You need to specifically tell Emacs that howm mode should be enabled in a particular document<sup>70</sup>. To do this, use the command: M-x home-mode

If you want Org documents to be friendly with howm by default and have all keyboard shortcuts work in them, add the following line to your .emacs file:

```
(add-hook 'org-mode-hook 'howm-mode)
```

On the plus side, howm will respond to commands from any Org document, regardless of its location. An even bigger plus is that howm links will work in any Org document (though you can only link to documents in the howm directory, since that's where the search is performed).

Be aware that these settings will affect the function of switching between buffers and closing buffers (see chapter “Working with Buffers”).

If you need to disable howm mode in some org document, use the same command we used to enable it: M-x home-mode



Howm uses many commands called action-lock. You are already familiar with the {} sign, which turns into today's date if you press RET inside the brackets. You can set up a similar “switch units” to perform other actions.

One switch unit is already placed in Howm as an example. Write {} in any note. Place the cursor inside and press RET. An asterisk will appear instead of a space - {\*}. Press RET again, and the character switches to a dash - {-}. Press RET again to return to a space.

You are free to prescribe any characters you want for this function. For example:

```
(setq action-lock-switch-default '("{" }" "{o}" "{o}" "{o}" "{o}" "{o}"))
```

You can use these symbols to indicate the completeness of notes or progress with tasks. To make it work correctly, you need to place this line in the .emacs file before the (require 'howm) line.

---

<sup>70</sup> Documents with the .org extension placed in the howm directory will function in both modes without any problems.



You can transform the start page of howm to your liking. For example, set the number of recent notes you want to see in the main menu (20 by default).

```
(setq howm-menu-recent-num 10)
```



You can change the look of the main Howm page by direct “surgery.” This is a regular `0000-00-00-000000.txt` file.

You can open it in any text editor. In the howm menu there is a special button that switches to the editing mode - [Edit Menu]. Clicking on it, you will see the following:

```
%"e" [Remember] %"c" [New] %"D" [Dup] Search(%"s" [String] %"g" [Regexp] %"o" [Occur])
%"a" [All] %"l" [Recent] %"A" [Around] %"y" [Schedule] %"t" [Todo] %"b" [Buffers] %"x" [Marks]
%"K" [<Title] [<Name] %"d" [Date] %"i" [Key] %"r" [Update] %"w" [Random Walk] [Killall]
%"." [Today] %"::" [Yesterday] %"h" [History] %"R" [Update Menu] [Edit Menu] [Preference]

[Schedule, Todo] — @schedule, !deadline (until %sdays days from now)
!deadline, +todo, -reminder, ~defer (top %tnum entries)
%reminder

Recent
%recent

Random -- If you do not like this, [Edit Menu] to erase it.
%random

Format of schedule and todo (Please replace {} with []):
{2002-10-21}@1 schedule -- (shown in schedule part. @3 = "3 days schedule to 10-23")
{2002-10-21}+7 todo -- float up slowly from the date in 7 days
{2002-10-21}!7 deadline -- float up fast from 7 days before the date
{2002-10-21}-1 reminder -- float at the date and sink slowly one unit per 1 day
{2002-10-21}~30 defer -- float at the date and repeat sinking and floating with 30 days period
{2002-10-21}. done -- sink forever
(Numbers after marks are default values.)
-:-- 0000-00-00-000000.txt Top L1 (Text Fly/— howm AL Wrap)
```

You can rearrange any elements in this file. For example, move the `%"h" [History]` button to the first place in the command section. Or swap places between recent and random notes. You can even remove an element altogether<sup>71</sup>.

You are also free to add additional items to the menu. It could be a link to some MOC note, a logon point.

Just write down the usual howm link somewhere:

>>> Index note

---

<sup>71</sup> Don't forget to back up this file before daring to experiment.

Another option is to add a search filter for a specific word. To do this, place somewhere the following entry:

```
%here%(howm-menu-search "your word")
```

This will show up as a list of notes in the menu.

You can, in a design frenzy, do something terribly beautiful to your homepage.

The screenshot shows a terminal window running Emacs. The title bar reads "\*howmM:0000-00-00-000000.txt\* - GNU Emacs at Andreis-MacBook-Pro.local". A blue bar at the top contains the text "%menu%". Below it is a large list of search commands in red and green. A horizontal line follows, then a section titled "[Schedule, Todo] -- @schedule, !deadline (until 7 days from now)" with several entries. Another horizontal line follows, then a decorative quote from Sirach 21:16: "The knowledge of the wise will increase like a flood..." with vertical lines and symbols around the text. The bottom status bar shows "U:%%- \*howmM:0000-00-00-000000.txt\* Top L13 (HM howm AL)".

But it's best to stick to minimalism.

# QUESTIONS AND CONCERNS

Q: Will howm slow down when I have numerous notes?

A: It depends on your operating system, Emacs version, and computer performance. With a few thousand notes, you shouldn't experience any noticeable difficulties. On my 2015 MacBook with Emacs 29.1, howm slows down in a test with 10,000 notes. It takes about 25 seconds to load the main menu or the full list of notes. This doesn't really concern me. Working with a knowledge base is a leisurely process. Besides, I have to live up to having 10,000 actually created notes.

Q: Will important notes get lost if I don't have a detailed structure?

A: The system enables important notes to stay prominent. Conversely, less interesting notes are pushed to the periphery. A certain chaos in the knowledge base proves to be a winning strategy. Conversely, overly rigid classification and structure burden the user with the constant need to tidy everything up.

Q: Is howm more efficient than other knowledge management systems?

A: It depends on your needs and preferences. There are no features in howm that a user couldn't replicate in other applications if desired. The advantage of howm lies in a reasonable balance between simplicity and features. Another advantage is its integration into the Emacs ecosystem. If you prefer working with Plain Text files, howm is a sensible choice.

# THE LIST OF MAIN HOWM COMMANDS

The meaning of commands becomes apparent within the context of working with notes, so consider this list a crib sheet. The last column contains page references for detailed discussions of each specific command. This table contains only shortcuts specific to the howm package; I skipped other useful Emacs shortcuts.

Shortcut	Command	Action	P
C-c , ,	M-x howm-menu	Call the main menu	11
C-c , c	M-x howm-create	Create a new memo	15
C-c , e	M-x howm-remember	Create a quick note	55
C-c , I	M-x howm-create-interactivly	Select a note name and localization	58
C-c , D	M-x howm-dup	Duplicate note	57
C-c , a	M-x howm-list-all	Call up a list of all notes	19
C-c , l	M-x howm-list-recent	Call up a list of recent notes	22
C-c , .	M-x howm-find-today	Call up a list of today's notes	22
C-c , :	M-x howm-find-yesterday	Call up a list of yesterday's notes	23
C-c , A	M-x howm-list-around	Show the note in temporal context	22
C-c , K	M-x howm-keyword-to-kill-ring	Copy a note title	28
C-u C-c , K	-----	Copy a numeric filename	29
C-c , s	M-x howm-list-grep-fixed	Search/recall keywords	38
C-c , g	M-x howm-list-grep	Search including regexp	38
C-c , o	M-x howm-occur	Search inside one note	52
C-c , h	M-x howm-history	Show search history	40
C-c , i	M-x howm-insert-keyword	Insert a keyword into a note	35
C-c , p	M-x action-lock-goto-previous-link	Jump to the previous link	32
C-c , n	M-x action-lock-goto-next-link	Jump to the next link	32
C-c , x	M-x howm-list-mark-ring	Show the list of marks	52
C-c , b	M-x howm-list-buffers	Call a list of open files	53
C-c , Q	M-x howm-kill-all	Close all buffers (except unsaved)	54
C-c , SPC	M-x howm-toggle-buffer	Switch between howm/non-howm files	53
C-c , t	M-x howm-list-todo	Call up the todo list	62
C-c , T	M-x howm-insert-dtime	Insert the current date and time	61
C-c , y	M-x howm-list-schedule	Call up a schedule	62
C-c , d	M-x howm-insert-date	Insert the current date	61
C-c , C	M-x howm-create-here	Create a heading with date in a note	51
C-c , N	M-x howm-next-memo	Jump to the next heading in a long note	50

C-c , P	M-x howm-previous-memo	Jump to the previous heading in a long note	50
C-c , L	M-x howm-last-memo	Jump to the last heading in a long note	50
C-c , H	M-x howm-first-memo	Jump to the first heading in a long note	50
C-c , r	M-x howm-refresh	Activate links on keywords in text	34
C-c , R	M-x howm-menu-refresh	Updating the main menu	—
C-c , M	M-x howm-open-named-file	Open a file (without howm mode)	58
C-c , w	M-x howm-random-walk	Random scrolling through notes	14